Identifying Significant Behaviour in Complex Bipartite Networks

A thesis submitted in fulfilment of the requirements for the degree of Doctor of Philosophy

by

Jessica Liebig
Bachelor of Science (Mathematics) (Honours), RMIT University

School of Science - Mathematical and Geospatial Sciences
College of Science, Engineering and Health
RMIT University

September 2016

Declaration of Authorship

I certify that except where due acknowledgement has been made, the work is that of the author alone; the work has not been submitted previously, in whole or in part, to qualify for any other academic award; the content of the thesis is the result of work which has been carried out since the official commencement date of the approved research program; any editorial work, paid or unpaid, carried out by a third party is acknowledged; and, ethics procedures and guidelines have been followed.

Jessica Liebig

September 12, 2016

Für Mama und Papa. Vor rund 16 Jahren fordertet Ihr mich dazu heraus eine Eins in Mathe zu schreiben. Danke, dass Ihr immer an mich glaubt und für mich da seid.

To my husband, who always finds the right words to make me laugh. Thank you for your constant support and for being incredibly patient.

"This question is so banal, but seemed to me worthy of attention in that [neither] geometry, nor algebra, nor even the art of counting was sufficient to solve it."

Leonhard Euler

Acknowledgements

First and foremost I wish to thank my senior supervisor Professor Asha Rao. Thank you Asha for your enormous support throughout my PhD candidature. During the almost five years that I have known you, you have become a mentor, a wonderful friend and a second mother to me. You have been there to listen to my worries and fears, regardless of whether they were research related or personal. You have put an incredible amount of work into supervising me as you do for any of your students. You read every single word that I wrote during my candidature at least three times. I also wish to thank you for the scholarship that you provided from your onwn research funding. You have gone above and beyond to guide and support me. Without your belief and motivation I could not have accomplished this thesis. Asha, you are a wonderful person and I could not have hoped for a better supervisor. I am incredibly grateful for the many things you taught me and will always look up to you.

I wish to thank my second supervisor Professor Kathy Horadam who encouraged me to apply for the information security pre-honours scholarship without which I would not have learned how much fun research brings. Kathy, you are one of the most amazing and enthusiastic teachers that I met during my Bachelor degree and you have become an inspiring role model. I am so very grateful for all the support and encouragement that you have given me during the years. Thank you Kathy for taking the time to read my thesis in a time that must be very difficult for you.

I also wish to thank Dr Stephen Davis, who listened to many of my presentations. Stephen, you attend all my milestone seminars and were happy to listen to many of my practice presentations prior to conferences. I very much appreciate you always taking the time to answer my questions and the valuable feedback you provided.

Thank you to all the staff members in the School of Mathematical and Geospatial Sciences for a truly enoyable work environment. Every single day in the department has been an absolute pleasure. I also wish to acknowlede the financial supprot, the fee waiver and funding for conferences, from RMIT and the School of Mathematical and Geospatial Sciences.

Thanks to all my friends at RMIT and outside university. Thanks to Rudaba, Solmaz and David with all of whom I have shared an office for the last eight months. You have made the final months of my PhD candidature a lot of fun. Thank you for taking me with you to all those lovely cafes to get coffee and sweet treats. All the laughter that we shared made these past few months a lot less stressful.

Thank you Shrupa for the many encouraging words. Your unique giggles are truly infectious. I sincerely enjoyed our many conversations, mostly over lunch. I am very glad to have you as a friend.

Elena, you are my oldest and best friend in Australia. I still remember the day we met at the English course that was meant to prepare us for the degrees that we were about to commence. We have known each other for more than eight years and a lot has changed during the years but our friendship has remained as strong as ever. Thank you for being there for me, and all your reassuring words.

A huge thank you to my best friend back in Germany. Danke liebe Hanni, dass Du immer für mich da bist, obwohl wir soweit voneinander weg sind. Danke für alle Deine Emails und Anrufe. Es fühlt sich an wie gestern als wir uns in der Grundschule kennengelernt haben. Seit dem sind wir unzertrennlich durch Dick und Dünn gegangen und haben jedes Ereignis miteinander geteilt. Ich vermisse unsere Freitag Nachmittage an denen wir zusammen mit Justus, Peter und Bob kniffelige Fälle gelöst haben und heisse Schokolade getrunken haben. Ich bin unglaublich dankbar Dich zu kennen und Dich als meine beste Freundin zu haben.

Danke liebe Joana. Als wir uns mit ungefähr zwölf Jahren kennenlernten, hätte ich nie gedacht, dass wir einmal im Mathe Leistungskurs sitzen werden und zusammen für Klausuren lernen. Es ist immer schön von Dir zu hören und sich an die tolle Zeit zu erinnern die wie miteinander von der siebten bis dreizehnten Klasse verbracht haben.

A very special thank you goes to my wonderful husband Urvik Bhalani for his immense support and unconditional love. Babu, you have been incredibly patient and I cannot describe how grateful I am. Thank you for your encouragement, your positive attitude and giving me the energy to write this thesis. You are there to give me a hug when I am crying and to tell me to believe in myself when I feel like giving up. Thank you for standing by my side and experiencing this important part of my life with me.

I wish to thank my husband's family for showing so much interest in what I am doing. Thank you Yogheshbhai, Chetnaben, Darshan and Zalak for treating me like your own daughter/sister. Thank you for the many calls and WhatsApp messages. Jay swaminarayan!

Lastly and most importantly, I want to thank my family. Liebe Mama, lieber Papa, lieber Nils, ich möchte Euch von ganzem Herzen danken. Ihr seid der Grund für all die Dinge, die ich erreicht habe. Ohne Eure Unterstützung wäre ich nie in der Lage gewesen diese Doktorarbeit zu schreiben. Ihr habt immer ohne jeden Zweifel an mich geglaubt und mich in jedem erdenklichen Weg unterstützt. Mama und Papa, Ihr seid die besten

Eltern die es gibt und obwohl ich am anderen Ende der Welt bin, seid Ihr immer ganz nah in meinem Herzen.

Contents

List of Figures

List of Tables

List of Arising Publications

LIEBIG, J. and RAO, A. (2016). Fast extraction of the backbone of projected bipartite networks to aid community detection. *Europhysics Letters, 113*: 28003.
DOI: 10.1209/0295-5075/113/28003

LIEBIG, J. and RAO, A. (2016). Predicting item popularity: Analysing local clustering behaviour of users. *Physica A, 442*: 523-531.
DOI: 10.1016/j.physa.2015.08.045

LIEBIG, J. and RAO, A. (2016). The case study of an Australian crime dataset. In: *Proceedings of the 3rd Annual Conference of Research@Locate*, 30-35.

LIEBIG, J. and RAO, A. (2014). Identifying influential nodes in bipartite networks using the clustering coefficient. In: *Proceedings of the 2014 Tenth International Conference on Signal-Image Technology and Internet-Based Systems*, 323-330.
DOI: 10.1109/SITIS.2014.15

Abstract

Identifying Significant Behaviour in Complex Bipartite Networks

The study of complex networks has received much attention over the past few decades, presenting a simple, yet efficient means of modelling and understanding complex systems. Networks are employed in various different areas, for instance, in the modelling of disease spread in human and animal contact networks. Networks also find applications in marketing, where various measures are used to recommend items to customers of, for instance, online shopping portals. Many other real world phenomena can be described and analysed using complex networks.

Most scientific literature focuses on the analysis of, so called, one-mode networks. However, many systems are best represented as bipartite networks. A network is bipartite if its vertices can be partitioned into two disjoint sets, where interaction takes place solely between vertices belonging to different sets. For instance, the network of scientists and papers, resulting from collaborations, is bipartite, with connections only existing between authors and papers. Similarly, the network of actors and the movies in which they appear is bipartite.

This thesis is motivated by the lack of network measures designed particularly for the analysis of bipartite networks. Since many one-mode network measures are not applicable to bipartite structures, often the only available path to analysing bipartite data is the examination of its projections. A projection converts a bipartite network into an ordinary one-mode network, causing loss of valuable information amongst other problems.

We are interested in both the theoretical aspects of bipartite networks and the applications to real world data. Throughout this thesis we analyse several real world networks with the aim of uncovering significant behaviour. We take two different approaches to gain a better understanding of complex bipartite networks. First, we deal with the problems that arise from the projection of bipartite networks, with the aim of overcoming these. Second, we develop network measures that are designed especially for bipartite networks.

Despite the many problems that arise from converting a bipartite network into a one-mode network, the study of projections is ubiquitous throughout the network science literature and projections are often preferred above the direct analysis of bipartite networks. The one-mode projection of a bipartite network is constructed by dropping one of its node sets and connecting two nodes of the remaining set if they share at least one neighbour in the bipartite network, leading to an inflation of edges in the projection. Furthermore, the indirect inference of edges between nodes in the one-mode projection leads to noise, that is, many edges with insignificant meaning are introduced. We develop a novel technique of identifying the significant connections that form the backbone of one-mode projections by considering the degree distributions of the bipartite network. We show that this identification of significant edges cannot be achieved by trivial methods such as an application of a threshold to the edge weights. Furthermore, we show that the weights of one-mode projections of real world bipartite networks follow a Poisson binomial distribution.

Real world one-mode projections often have well hidden community structures. These structures can be uncovered by dropping insignificant connections, as identified by our technique. In addition, our technique allows a ranking of edges by importance. We apply this backbone technique to three different real world networks, and show that our method is a very efficient way of identifying communities within diverse networks, such as the political parties in a Facebook network of posts by candidates and user likes.

The development of new network measures that can be applied directly to bipartite networks is a crucial step towards a better understanding of these structures. One of the most important and widely used network measures is the clustering coefficient. Due to the particular structure of bipartite networks, the clustering coefficient cannot be directly applied to them. Although several definitions for the bipartite clustering coefficient have been presented in the literature, they are inconsistent and hence we explore this topic in great depth. We identify different types of bipartite networks based on their development over time, consequently requiring different definitions of the bipartite clustering coefficient. We precisely define the different types of networks before providing new definitions of the clustering coefficients for each type of bipartite network.

We apply our clustering coefficients to discover the most influential nodes in real world bipartite networks by introducing the notion of the driving score. The driving score indicates the extent to which each individual node contributes to the overall clustering behaviour of the network. Another application of our clustering coefficient is the prediction of the future popularity of new items in rating networks. We are able to considerably improve existing predictions.

Crime networks form a very interesting group of bipartite networks. Knowledge about their dynamics is especially important for the implementation of efficient crime prevention measures. We present two case studies of crime networks revealing many interesting insights, by using a combination of both the approaches outlined above. For instance, our analysis reveals significant co-occurrences of illegal activity and identifies areas that exhibit similar crime dynamics.

The calculation of many network measures, including the ones we introduce in this thesis, require the enumeration of sub-graphs. In the last chapter of this thesis, we investigate several efficient ways of enumerating sub-graphs in bipartite networks, by studying, combining and modifying existing algorithms. We also present preliminary work on the theoretical problem of path enumeration.

Chapter 1

Introduction

1.1 Motivation

The study of complex networks has received much attention in recent years, leading to the discovery of many interesting and sometimes surprising results. For example, the study of friendship networks has shown that your friends, on average have more friends than you do [36]. In 1967 Milgram [75] experimentally showed that on average any two people on earth are only six steps away from each other. Here, steps are a chain of acquaintances.

Complex networks are mathematical structures that are employed to model and understand complex systems in varied areas of real life. An example is the modelling of the spread of disease through contact networks, while another is the recommendation of items to users of online shopping portals.

Researchers mainly concentrate on the analysis of one-mode networks, networks with only one type of nodes. There are however many real world systems that consist of two or more different types of entities. Systems with two types of entities are best modelled by bipartite networks that are particularly structured networks comprising of two different sets of nodes, with connections only between nodes belonging to different sets.

The motivation for this thesis lies in the lack of network measures tailored particularly for bipartite networks. Due to this lack bipartite networks are generally not analysed directly, but by projection onto a one-mode network. One-mode projections are simplifications of bipartite networks, that only contain one of the two node sets. Two nodes in the remaining set are connected if they share at least one neighbour in the bipartite network. While this process of projecting a bipartite network allows the application of one-mode network measures, it also causes many problems.

In this thesis we tackle some of the problems that arise in the analysis of bipartite networks, first by developing a technique that overcomes some limitations of the one-mode projection, and second, by developing measures to facilitate the direct analysis of bipartite networks.

1.2 Scope and contribution

The aim of this thesis is the identification of significant behaviour in bipartite networks. We develop network measures and techniques specifically designed for the analysis of bipartite networks. In particular, our contributions to the literature are the following:

- We demonstrate that the edge weights of a projected bipartite network follow a Poisson binomial distribution.
- We use the above result to introduce a novel technique for extracting the significant edges of one-mode projections and demonstrate that elimination of insignificant connections reveals the community structure within the projection.
- We formally show that the global one-mode clustering coefficient of a projected bipartite network is generally higher than that of a similar random network.
- We define two different types of bipartite networks that differ in the way they develop over time.
- For each of the identified types of bipartite networks we define suitable bipartite clustering coefficients.
- We apply the bipartite clustering coefficients to detect the most important nodes within real world bipartite networks by introducing the concept of the driving score.
- We apply the clustering coefficients to predict the popularity of new items in rating networks.

- We present two case studies of crime networks, revealing interesting insights into their dynamics and raising several questions that will be addressed in future work.
- We demonstrate that the occurrence of particular sub-graphs in burglary event networks is biased.
- We present preliminary results on the enumeration of sub-graphs in bipartite networks.

1.3 Structure of the document

This thesis is divided into eight chapters. The current chapter states the motivation, scope, and contribution of this thesis.

Chapter 2 provides the reader with the necessary background in network science. It contains the relevant definitions, notation, and preliminaries needed in the remainder of the thesis.

Chapter 3 critically studies the process of projecting a bipartite network onto a one-mode network. Our attention is drawn to one particular problem that arises during this process, that of edge inflation. Not every edge in the one-mode projection of a bipartite network has significance, due to the indirect inference of connections between nodes. Thus, projections are not only dense, but noisy networks. By demonstrating that the weights of one-mode projections follow a Poisson binomial distribution, we identify the statistically most significant edges in a fast and efficient manner. Furthermore, we show that deletion of insignificant edges leads to a well pronounced community structure within the one-mode projection.

Chapters 4 and 5 are dedicated to the bipartite clustering coefficient. While Chapter 4 discusses the theoretical aspects, several applications to real world data are studied in detail in Chapter 5.

The clustering coefficient is an important measure that has led to many useful results in the analysis of one-mode networks. Due to the special structure of bipartite networks, the one-mode clustering coefficient of any bipartite network is zero, making the measure meaningless. It is well known amongst network scientists that the clustering coefficient of one-mode projections is generally much higher than that of similar random one-mode networks. We formally show that this is indeed the case.

Several definitions for the bipartite clustering coefficient have been presented in the literature. However, they are inconsistent. After carefully reviewing the existing bipartite clustering coefficients we identify two major limitations. First, it is important to understand that different bipartite networks develop differently over time, affecting the formation of clusters. We precisely define the different types of bipartite networks. Second, in a bipartite network clusters can have different structures. We clearly identify these structures and give a mathematical proof that ignoring these structures will lead to inaccurate results. We then introduce different bipartite clustering coefficients, one for each type of network, thus overcoming the existing limitations.

In Chapter 5 we look at several applications of our novel bipartite clustering coefficients. We demonstrate that it can be used to identify the most important and influential nodes of a network. We introduce a novel measure, called the driving score, that utilises the bipartite clustering coefficient to rank the nodes of a given network by importance. Another application of our clustering coefficient is the prediction of future item popularity in rating networks. We establish a novel technique that drastically improves on existing methods.

Chapter 6 is devoted to the analysis of crime networks. The analysis of crime data is a necessary step towards the prevention of criminal activity. We present two case studies in which we combine the measures and techniques we developed in earlier chapters to demonstrate their potential. The work presented in Chapter 6 is ongoing work. We raise several questions that we will address in future research. In addition, we show that the occurrence of particular sub-graphs that were identified as motifs by an earlier study, is biased.

Chapter 7 presents preliminary work on the enumeration of sub-graphs in bipartite networks. We describe two of the fastest algorithms used for the detection of motifs, sub-graphs that occur with a significant frequency in an observed network. We modify one of the algorithms to enumerate the sub-graphs needed for the calculation of the bipartite clustering coefficients that we introduce in Chapter 4. The remainder of Chapter 7 presents preliminary results on the enumeration of paths on the square lattice.

Chapter 2

Background

2.1 Introduction

The area of network science has developed rapidly over the past decade. This chapter serves as a point of reference, providing definitions, notations and other preliminaries used and referred to in the remainder of this thesis. In addition, every other chapter has its own introductory section, giving its motivation and providing relevant information.

All notation and definitions in this chapter and the thesis as a whole follow as far as possible conventions from graph theory. Consequently some definitions may be different in networks science.

This chapter is structured as follows: In Section 2.2 we give the relevant background in graph theory and network science by providing definitions, notation and the required preliminaries. Section 2.3 presents material on probability distributions and generating functions that will be used to derive theoretical results about networks in later chapters. Section 2.4 introduces the datasets that will be studied in later chapters.

2.2 Network science

The study of complex networks and the theory of graphs are closely related. Graphs have been studied long before the emergence of network science, dating as far back as the 18^{th} century when Leonhard Euler solved the Bridges of Königsberg problem. In

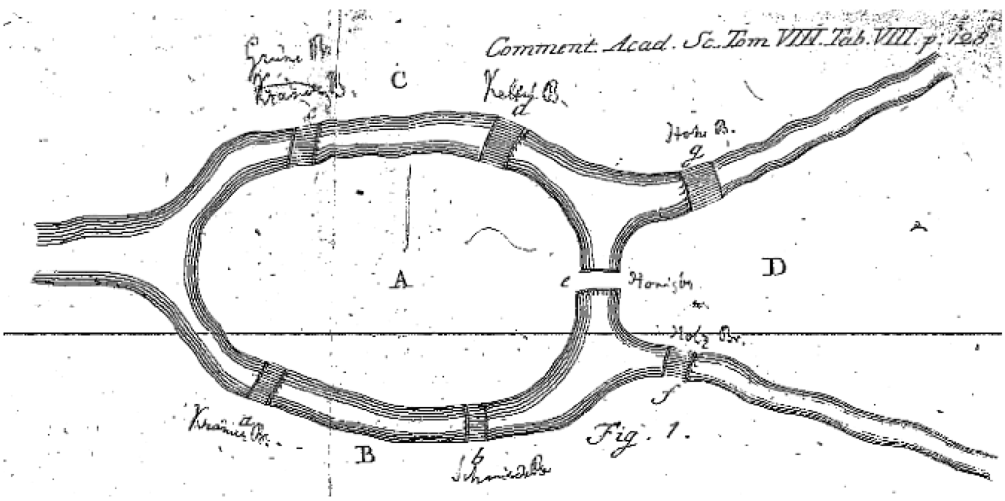


FIGURE 2.1: Euler's map of Königsberg as shown in his 1741 paper. The seven bridges are labelled a, b, c, d, e, f and g, the four landmasses are labelled A, B, C and D.

1741 Euler published a paper, proving that it is impossible to find a path that crosses each of the seven bridges of Königsberg exactly once [32]. Figure 2.1 shows Euler's map of Königsberg. In his paper, Euler states his belief of the problem being related to the geometry of positions, today known as graph theory [96].

The map depicted in Figure 2.1 can be represented as a graph by assigning a node to each of the four landmasses. A pair of nodes is connected by an edge if the two corresponding landmasses are connected by a bridge (see Figure 2.2).

Following we give relevant definitions and notation all of which can be found in standard textbooks on graph theory and network science [e.g. 28, 86].

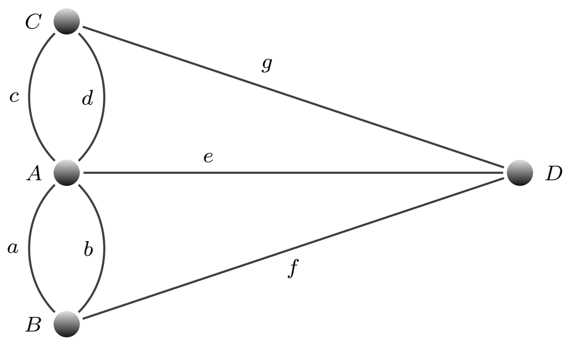


FIGURE 2.2: The graph representing Euler's map of Königsberg depicted in Figure 2.1.

2.2.1 Definitions and notation

The two terms *graph* and *network* are often used interchangeably in the network science literature. This subsection introduces terms, definitions and notation taken from graph theory.

A graph consists of *nodes* and *edges*, also called *vertices* and *links*. The edges of a graph connect pairs of nodes to each other and may be *directed* or *undirected*, indicating a certain type of relationship between the nodes. Edges often have attributes such as *weights*, representing for example the strength of connection between two nodes. Figure 2.3 shows an example of a graph, consisting of 23 nodes and 24 undirected edges.

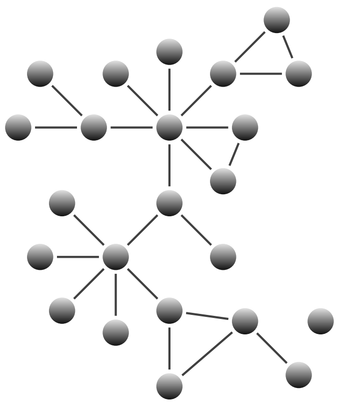


FIGURE 2.3: An undirected graph of order 23 and size 24. One of its nodes is isolated and hence, the graph is not connected.

Definition 2.1. A *graph* is a pair $\mathcal{G} = (U, E)$, where U is the set of nodes and E is the set of edges. The edge $e_{ij} = (u_i, u_j) \in E$, if it exists, connects node u_i to node u_j. In an *undirected* graph, $e_{ij} \in E \Leftrightarrow e_{ji} \in E$.

Definition 2.2. A graph is *bipartite* if its set of vertices can be partitioned into two disjoint sets, U and V, such that $U \cap V = \emptyset$ and $E \subseteq U \times V$. A bipartite graph is denoted $\mathcal{B} = (U, V, E)$, where U is called the *primary node set* and V is called the *secondary node set*.

In graph theory, the *order* of a graph $\mathcal{G}$ is defined as the number of its nodes, whereas the *size* of $\mathcal{G}$ is defined as the number of its edges. In network science, the size of a network sometimes refers to the number of its nodes. We choose to follow the graph theoretical terminology and hence call the number of nodes in network $\mathcal{G}$ its order, denoted by $|\mathcal{G}| = |U|$. For a bipartite network $\mathcal{B}$, $|\mathcal{B}| = |U| + |V|$. The size of a network is given by the number of its edges and is denoted $||\mathcal{G}|| = |E|$.

Definition 2.3. The edge $e_{ii} = (u_i, u_i)$ connecting node u_i to itself is called a *loop*.

Definition 2.4. A graph is *simple* if it does not contain any loops or multiple edges.

Definition 2.5. The simple graph $\mathcal{G} = (U, E)$ is *complete* if all nodes in U are pairwise adjacent. The complete graph of order n is denoted $\mathcal{K}_n$. The bipartite graph $\mathcal{B} = (U, V, E)$ is complete if $E = U \times V$ and is denoted $\mathcal{K}_{m,n}$, where $|U| = m$ and $|V| = n$.

Definition 2.6. $\mathcal{G}' = (U', E')$ is a *sub-graph* of $\mathcal{G} = (U, E)$, if $U' \subseteq U$ and $E' \subseteq E$. $\mathcal{G}'$ is an *induced sub-graph* of $\mathcal{G}$ if $E' = \{e_{ij} \mid e_{ij} \in E,\ \forall u_i, u_j \in U'\}$.

Definition 2.7. A complete sub-graph is called a *clique*.

Definition 2.8. A sub-graph of order $n+1$ with n nodes having degree one and one node having degree n is called an *n-star*.

Figure 2.4 shows two examples of star-sub-graphs.

(A) (B)

Figure 2.4: A 3-star (A) and a 4-star (B).

Definition 2.9. Two graphs $\mathcal{G} = (U, E)$ and $\mathcal{H} = (U', E')$ are isomorphic, denoted by $\mathcal{G} \simeq \mathcal{H}$, if the exists a bijection $\varphi : U \to U'$ such that $(u_i, u_j) \in E \Leftrightarrow (\varphi(u_i), \varphi(u_j)) \in E',\ \forall u_i, u_j \in U$.

Definition 2.10. The *degree* of node u is equal to the number of its adjacent edges and denoted by $\deg(u) = j_u$. The *average degree* over all nodes in a graph $\mathcal{G} = (U, E)$ is denoted by

$$\langle j \rangle = \frac{1}{|U|} \sum_{u=1}^{|U|} j_u. \tag{2.1}$$

Note that the sum of the degrees of a graph is equal to twice the number of its edges.

In a bipartite graph, we denote the average degree of the primary node set by $\langle j \rangle$ and the average degree of the secondary node set by $\langle k \rangle$.

A node of degree zero is an *isolated* node. If all vertices in graph $\mathcal{G}$ have equal degree, i.e. $\deg(u_i) = k,\ \forall u_i \in U$, then $\mathcal{G}$ is said to be k-regular. In the network science literature, a k-regular graph is often called a strictly homogeneous network.

Two nodes that are connected by an edge are called *adjacent* nodes or *neighbours*. Nodes can be connected via *walks*, *trails* and *paths*.

Definition 2.11. A *walk* on the graph $\mathcal{G}$ is a sequence of nodes and edges, where nodes and edges do not have to be unique. The n-walk W_n of length n from node u_0 to node u_n is written $W_n = u_0 u_1 \dots u_n$, listing all the nodes that it visits in order.

Definition 2.12. A *trail* is a walk, with all edges being unique. Nodes may be visited multiple times. The n-trail T_n is written $T_n = u_0 u_1 \dots u_n$.

Definition 2.13. A *path* is a walk, with all its elements, nodes and edges, being unique. The n-path P_n is written $P_n = u_0 u_1 \dots u_n$.

Definition 2.14. A *cycle* or *circuit* is a path that starts and ends at the same vertex. All other elements must be unique. The n-cycle C_n is written $C_n = u_0 \dots u_{n-1} u_0$.

Note that in the literature the term path is often used to mean a walk. We use the terms walk, trail and path as defined above.

A graph is *connected* if there exists a path between any two vertices of the graph.

Definition 2.15. An edge connecting two nodes that are part of a cycle and that itself is not part of the cycle, is called a *chord*. A cycle without any chords is an *induced cycle*.

Definition 2.16. The *distance* between two vertices u_i and u_j is the length of the shortest path (geodesic) between them and denoted by $d(u_i, u_j)$.

2.2.2 Matrix representations of networks

Graphs and networks are commonly represented in form of their adjacency matrices.

Definition 2.17. Let $\mathcal{G} = (U, E)$ be a graph of order n. Its *adjacency matrix* A is the binary $n \times n$ matrix with elements a_{ij}, such that $a_{ij} = 1$ if $(u_i, u_j) \in E$ and 0 otherwise. If $\mathcal{G}$ is undirected $a_{ij} = a_{ji}$.

The n^{th} power A^n of the adjacency matrix contains as its entries the number of walks of length n between nodes u_i and u_j.

If A is the adjacency matrix of some bipartite graph, then there exists a matrix A' that is isomorphic to A such that:

$$A' = \begin{bmatrix} \mathbf{0} & B \\ B^T & \mathbf{0} \end{bmatrix}, \tag{2.2}$$

where $\mathbf{0}$ is the all zero matrix, B is the biadjacency matrix (see Definition 2.18) and B^T is its transpose.

Definition 2.18. Let $\mathcal{B} = (U, V, E)$ be a bipartite graph. Its *biadjacency matrix* B is the binary $|U| \times |V|$ matrix with elements b_{ij}, such that $b_{ij} = 1$ if $(u_i, v_j) \in E$ and 0 otherwise.

2.2.3 Network measures

There exist a vast number of measures and metrics for the analysis of complex networks. This subsection reviews the most basic measures, all of which can be applied to bipartite networks with some slight modifications [14]. More complex measures are introduced in later chapters where needed.

2.2.3.1 Network density

Density measures the edge density of a network and is given by $|E|/(|U|(|U|-1))$, where $|U|(|U|-1)$ is the maximum possible number of edges in a simple network with $|U|$ nodes.

In a bipartite network, two nodes of the same type cannot be connected and hence, when calculating the density of a bipartite network the denominator must be rewritten as $|U||V|$. Therefore, the density of a bipartite network is given by $|E|/(|U||V|)$.

2.2.3.2 Degree centrality

The *degree centrality* is one of the simplest of the network measures. The degree centrality of a node is equal to its degree. The nodes with the highest degree centralities are called *hubs*. This measure can be applied to bipartite networks without any modifications.

2.2.3.3 Betweenness centrality

The *betweenness centrality* of a node is a more complex measure of how centrally a node is positioned in a network. Betweenness centrality measures the number of shortest paths that run through a given node. The betweenness centrality of node u_i is given by

$$b_i = \sum_{j,k} \frac{g^i_{jk}}{g_{jk}}, \tag{2.3}$$

where g_{jk} is the number of shortest paths between nodes u_j and u_k and g^i_{jk} is the number of shortest paths between nodes u_j and u_k that contain node u_i. If $g_{jk} = g^i_{jk} = 0$, $g^i_{jk}/g_{jk} = 0$ by definition.

2.2.3.4 Closeness centrality

Closeness centrality is another popular measure of centrality that calculates the inverse of the average distance of a node to every other node in the network. The closeness centrality of node u_i is given by

$$c_i = (|U| - 1) \Bigg/ \sum_{j \neq i} d(u_i, u_j) \; . \tag{2.4}$$

In a bipartite network the minimum distance between two nodes of the same type is two, whereas the minimum distance between two nodes of different type is one. This has to be taken into consideration when calculating the closeness of a node in a bipartite network [14].

There are a number of other network measures that we will define in later chapters as needed.

2.2.4 Network communities

In Chapter 3 we introduce a novel and efficient way of identifying significant connections in one-mode projections, building an aid to detecting network communities. A network community is loosely defined as a group of nodes that is well connected. In other words, the density of edges within a community is relatively higher than the density of edges between communities. Many real world networks exhibit community structure [42]. Figure 2.5 shows a small network with three communities.

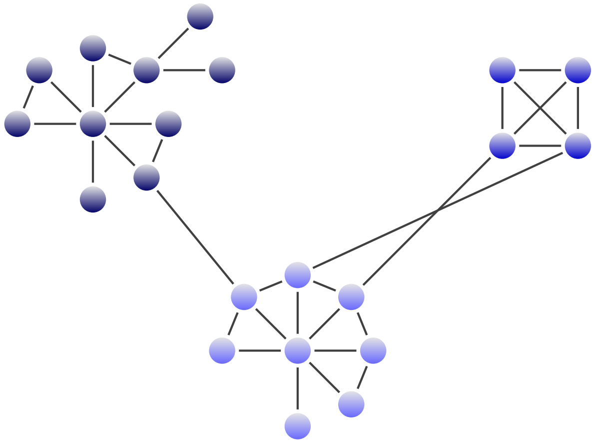


FIGURE 2.5: A small network with community structure. Nodes are coloured according to their community membership.

The network science literature pays much attention to the problem of detecting the communities of a network [23, 79, 88, 99, 104, 110]. Their identification is a challenging task, as communities can overlap or may be well hidden within the network structure [25].

Many community detection algorithms have been developed, using different approaches. Some are based on modularity, while others use spectral methods or probability theory. Early detection algorithms required the user to input the number of network communities, usually unknown prior to analysis. Algorithms have been improved to overcome this limitation.

Before outlining a few popular community detection algorithms that we will apply in Chapter 3, we look at the modularity function. Modularity measures the quality of a particular division of a network into groups of nodes found by a community detection algorithm [87].

2.2.4.1 The modularity function

The modularity of a particular division of a network into groups of nodes can be calculated by subtracting the number of expected edges within these groups if the network was random, from the number of observed edges within the groups. Large values of modularity indicate a well pronounced community structure in the observed network. Note that random networks are said to have no community structure.

To calculate the expected number of edges within the different groups of vertices one needs an ensemble of random networks similar to the one under investigation for comparison purposes. There exist many techniques to generate the ensemble and we outline the configuration model, one of the most popular models in Subsection 2.2.5.1. Once the preferred random network model is chosen, one can calculate the probability of observing an edge between any two nodes of the network and then subtract this number from the observed number of edges in the network under consideration. Newman [85] formally defines the modularity as

$$Q = \frac{1}{2m} \sum_{i,j} [a_{ij} - p_{ij}] \, \delta(g_i, g_j), \tag{2.5}$$

where $2m$ is the sum of the degrees in the network, a_{ij} is the $(ij)^{\text{th}}$ entry of the adjacency matrix A, p_{ij} is the expected number of edges between nodes u_i and u_j if the network was random, g_i is the community with which node u_i is associated and $\delta(a, b) = 1$ if $a = b$ and 0 otherwise.

The modularity function allows the evaluation of the quality of a particular partition of a given network into groups of nodes. The higher the modularity, the higher the quality of the partition.

Following we summarise three community detection algorithms, all of which we will apply in Chapter 3. At this point we wish to remark that we are not interested in evaluating the performances of the different algorithms. The primary reason for choosing the below listed algorithms is that they have been implemented in the R programming language [24]. We are aware of the ongoing discussion about the performance of community detection algorithms and their individual limitations and we will discuss some of these below.

2.2.4.2 Louvain

Blondel et al. [12] introduced a fast algorithm that partitions large networks into communities with the aim of maximising the modularity function given by Equation (2.5). The algorithm begins with every node forming its own community. In the next step the algorithm iterates through the neighbours of each node and checks if the modularity of the network can be increased by merging the two communities.

Many community detection algorithms are based on maximising the modularity function (see [85] and the references therein). Despite this approach being very popular, several problems have been pointed out. For example, modularity maximisation tends to group small well separated clusters together while simultaneously dividing large groups of well connected nodes [59]. On the other hand, one of the greatest advantages of the Louvain algorithm is its almost linear (in the number of edges) computation time [58]. In 2009 Lancichinetti and Fortunato [58] carried out a comparison between several community detection algorithms, with the Louvain algorithm being one of the best performing algorithms. The algorithm with the best results was Infomap [110, 111]. A new version of Infomap is available online (`http://www.mapequation.org/`), however, this version has not been compared to other community detection algorithms and hence is not used here.

2.2.4.3 Leading eigenvector

Newman's [85] community detection algorithm, based on the leading eigenvector of the modularity matrix, aims to divide the input network into groups such that the modularity function (see Equation (2.5)) is maximised. Rewriting Equation (2.5) in terms of matrices allows one to view the optimisation problem as a spectral problem. As the community structure of a network is often encoded in the first few eigenvectors of the modularity matrix, the complexity of the initial optimisation problem is reduced. The advantage of this algorithm lies in the eigenvalues not being dependent on any particular division of the network into groups of nodes.

2.2.4.4 WalkTrap algorithm

Pons and Latapy [101] introduce a community detection algorithm based on random walks. Since the nodes within a community are densely connected while the communities are connected by relatively fewer edges, a random walk on a network with community structure is more likely to stay within a community than to traverse between communities.

The WalkTrap algorithm, similar to the Louvain algorithm and the leading eigenvector algorithm, aims to maximise the modularity function. However, it is very different in its approach as it makes use of random walks.

2.2.5 Random networks

Random networks are used as a means of comparison. Being able to compare a given network to an ensemble of random networks is important to determine whether the occurrence of a particular pattern in the network of interest is significant. Much effort has been invested into developing randomisation techniques, mainly for one-mode networks. Researchers agree that a random network that is used for comparison needs to have the same order and size, and the same degree distribution or sequence as the observed network.

2.2.5.1 The configuration model

The perhaps most widely studied random network model that fulfils the above conditions and that can be applied to one-mode and bipartite networks alike is the configuration model (see [84] and the references therein). The configuration model begins with a network of given order and size zero. Each node has a specified number of so-called stubs corresponding to its degree. Hence, the number of nodes in the random network and its degree sequence are fixed. Since the total number of stubs is equal to twice the number of edges, the number of edges is also fixed. Next, the model chooses two stubs at random and links them with an edge. This process is repeated until all stubs are connected (see Figure 2.6).

In the remainder of this thesis, we randomise observed networks by using a particular algorithm, the Curveball algorithm, that yields the same kind of networks as the configuration model. This algorithm is outlined in the next subsection.

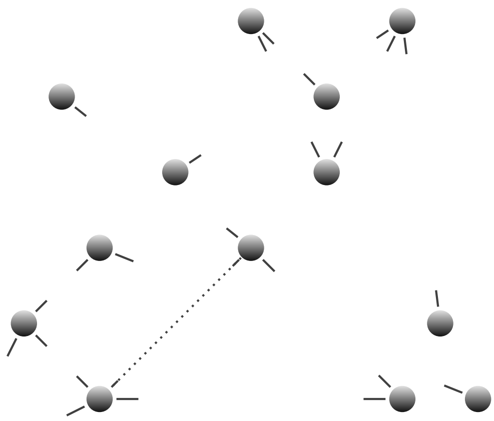


FIGURE 2.6: In each step of the configuration model two stubs are chosen at random and connected with an edge.

2.2.5.2 The Curveball algorithm

The Curveball algorithm [119] is similar to the well known switching model [76]. The switching model randomises a given network by randomly choosing two edges and then swapping the two nodes at the ends of the edges (see Figure 2.7). This procedure ensures that the order, the size and the degree sequence of the network are fixed.

FIGURE 2.7: One switch of the switching algorithm. Two edges are randomly chosen and the nodes at the ends (u_2 and u_4) are swapped.

The Curveball algorithm randomises the adjacency matrix of a network, fixing its row and column sums, in a similar manner to the switching algorithm. Fixing the row and column sums of the adjacency matrix is equivalent to fixing the degree sequence of the network. To randomise the adjacency matrix A, the Curveball algorithm randomly chooses two rows of A, say A_i and A_j. The two rows are compared by creating a list l_1 that holds the column indices that contain a one in A_i but not in A_j. A second list, l_2, holds the column indices that contain a one in A_j but not in A_i. Next, two new row vectors are created by removing all ones from A_i and A_j that have a column index that is contained in l_1 and l_2 respectively. The same number of ones that were removed from A_i are added at the positions of randomly chosen indices from $l_1 \cup l_2$. The remaining elements in $l_1 \cup l_2$ are added to A_j. These steps are repeated N times. Finally, a new matrix is formed from the resulting row vectors.

The Curveball algorithm has the advantage of being much faster than other switching algorithms, as it carries out multiple switches simultaneously. The primary reason for choosing the Curveball algorithm for our randomisations is its convergence to the uniform distribution. A proof of this convergence to the uniform distribution can be found in [18].

2.2.5.3 Finding bipartite graphic sequences

While there are many algorithms that randomise a given network, it is very challenging to create a random network with a given degree distribution from scratch since it involves finding a graphic or bipartite graphic sequence that follows a given distribution. In this subsection we provide a possible algorithm to create a random bipartite network, with its primary and secondary degrees following prescribed degree distributions.

Definition 2.19. A finite sequence $s = (s_1, s_2, \ldots, s_n)$ of non-negative integers is *graphic* if it is the degree sequence of a simple graph $\mathcal{G}$ of order n. $\mathcal{G}$ is called a *realisation* of s.

The sequence $s = (2, 3, 5, 6, 6, 9)$ for example is not graphic, since the sum of degrees of any graph needs to be even. Hence, for a non-negative sequence of integers to be graphic its sum has to be even. This is a necessary condition. Another necessary condition is that any element in the sequence has to be less than the length of the sequence, since in a simple graph of order n the maximum possible degree of a vertex is $n - 1$.

Definition 2.20. A pair of finite sequences (r, s) of non-negative integers, where $r = (r_1, r_2, \ldots, r_m)$ and $s = (s_1, s_2, \ldots, s_n)$, is *bipartite graphic* if the sequences r and s form the primary and secondary degree sequences of a simple bipartite graph of order $m + n$, respectively.

The two afore mentioned necessary conditions for a sequence to be graphic are also necessary for a pair of non-negative sequences of integers to be bipartite graphic. In addition $\sum_{i=1}^{m} r_i = \sum_{j=1}^{n} s_j$, that is, the sum of the primary degrees has to equal to the sum of the secondary degrees.

The Gale-Ryser Theorem gives a necessary and sufficient condition for a pair of non-negative sequences of integers to be bipartite graphic [1]:

Theorem 2.21 (Gale-Ryser Theorem). *The non-increasing pair of sequences (r, s) is bipartite graphic if and only if*

$$\sum_{i=1}^{m} r_i = \sum_{j=1}^{n} s_j \text{ and } \sum_{i=1}^{k} r_i \leq \sum_{j=1}^{n} min(s_j, k), \ \forall\ k = 1, \dots m.$$

Theorem 2.21 allows us to test whether a sequence drawn from a given distribution is bipartite graphic. On the other hand, finding a realisation of a bipartite graphic sequence is another problem. The proof of the following theorem will clarify the construction of a realisation of a bipartite graphic sequence [1].

Theorem 2.22. *A non-increasing pair of sequences (r, s) is bipartite graphic if and only if $r_1 \leq n, \sum_{i=1}^{m} r_i = \sum_{j=1}^{n} s_j$ and the pair (r', s') is also bipartite graphic, where $r' = (0, r_2, \dots, r_m)$ and $s' = (s_1 - 1, \dots, s_{r_1} - 1, s_{r_1+1}, \dots, s_n)$.*

Proof. Assume that the pair of sequences (r', s') is bipartite graphic with a realisation $\mathcal{B} = (U, V, E')$, where U has degree sequence r' and V has degree sequence s'. To construct a realisation of the pair (r, s) from $\mathcal{B}$, we can join the isolated node in U to the nodes in V having degrees $s_1 - 1, \dots, s_{r_1} - 1$. We remind the reader that $r' = (0, r_2, \dots, r_m)$ and hence at least one node that belongs to the primary node set is isolated.

Now assume that the pair of sequences (r, s) is bipartite graphic with some realisation $\mathcal{B} = (U, V, E)$ and let node $u_1 \in U$ be adjacent to nodes $v_1, \dots, v_{r_1} \in V$. A realisation of (r', s') can be constructed by deleting all edges between the nodes u_1 and $v_1, \dots, v_{r_1}$.

If node u_1 is not connected to one of the nodes in $\{v_1, \dots, v_{r_1}\}$, say v_i, then it must be connected to another node v_j, with $j > r_1$, and $\deg(v_j) \leq \deg(v_i)$ since the elements of r and s are placed in non-increasing order. Hence, there must be a node u_k that is connected to v_i but not to v_j. By switching the ends of the two edges (u_1, v_j) and (u_k, v_i), i.e. deleting (u_1, v_j) and (u_k, v_i) from $\mathcal{B}$ and adding the two edges (u_1, v_i) and (u_k, v_j), we connect node u_1 to node v_i. Repeating this step if necessary we can obtain a realisation of (r, s) with node u_1 being connected to nodes $v_1, \dots, v_{r_1}$. Hence we can find a realisation of (r', s'). □

The proof provides a possible way of finding a realisation of a pair of sequences (r, s) that is bipartite graphic:

First we order the elements of r and s in non-increasing order, such that $r = (r_1, r_2, \ldots, r_m)$, with $r_1 \geq r_2 \geq \cdots \geq r_m$ and $s = (s_1, s_2, \ldots, s_n)$ with $s_1 \geq s_2 \geq \cdots \geq s_n$. We start with the bipartite graph $\mathcal{B} = (U, V, E)$ of order $m + n$ and size zero. Let $\deg(u_1) = r_1$, where $u_1 \in U$ and r_1 is the maximal element of r. We join r_1 edges from node u_1 to nodes $v_1, \ldots, v_{r_1} \in V$. Next, consider the pair (r', s') and connect node u_2 to the first r_2 nodes in V for which $s'_i > 0$. Repeating this procedure will result in a realisation of (r, s) [1].

We have implemented these steps in the R programming language [103] to use in later chapters for the generation of random bipartite networks. Algorithm 1 displays the pseudo code of our program.

Algorithm 1 Creating one realisation of a bipartite graphic pair of sequences.

procedure ONEREALISATION(r, s)
    $m \leftarrow$ **length**(r)
    $n \leftarrow$ **length**(s)
    if $\sum_i r_i \neq \sum_j s_j$ **then**
        exit ▷ The sums of both degree sequences have to be equal.
    end if
    for $k \leftarrow 1,\ m$ **do**
        if $\sum_i^k r_i > \sum_j^n \min(s_j, k)$ **then**
            exit ▷ Not a bipartite graphic sequence.
        end if
    end for
    $r = (r_1, r_2, \ldots, r_m),\ r_1 \geq r_2 \geq \cdots \geq r_m$
    $s = (s_1, s_2, \ldots, s_n),\ s_1 \geq s_2 \geq \cdots \geq s_n$
    $U \leftarrow \{u_1, \ldots, u_m\}$
    $V \leftarrow \{v_1, \ldots, u_n\}$
    index $\leftarrow 1$
    for $i \leftarrow 1,\ m$ **do**
        if $s[\text{start}] = 0$ **then**
            index $\leftarrow$ index $+1$
        end if
        $E \leftarrow \{(u_i, v_{\text{index}}), (u_i, v_{\text{index}+1}), \ldots, (u_i, v_{\text{index}+r_i-1})\}$
        $r[i] \leftarrow 0$
        $s[\text{index} : (\text{index} + r_1 - 1)] \leftarrow s[\text{index} : (\text{index} + r_1 - 1)] - 1$
    end for
    return $\mathcal{B} = (U, V, E)$
end procedure

The final process of creating a random bipartite network with its primary and secondary degrees following prescribed distributions is as follows:

We draw two sequences of values from given distributions. We can slightly modify the two sequences to ensure that their sums are equal and then input them into Algorithm 1.

Since Algorithm 1 produces one particular realisation of a bipartite graphic pair of sequences that is not randomly created, we feed the realisation into the Curveball algorithm to produce a random bipartite network.

While generating random bipartite networks in this manner we noticed the longer the sequences, i.e. the greater the desired order of the random bipartite network, the likelier it is a pair of sequences that is bipartite graphic can be found. It would be worth exploring this notion in future research.

2.3 Probability distributions and generating functions

In Chapters 3 and 4 we will use the degree distributions of bipartite networks to deduce a number of interesting results. The degrees of real world networks are distributed in a certain manner and may be fitted to a particular probability function. All definitions and equations provided in this section can be found in standard textbooks on probability and generating functions [e.g. 38, 60].

2.3.1 Common probability distributions

The probability density function $P(X = x)$ of a random variable X gives the probability that X takes value x. An example of a random variable is the degree of a randomly chosen node. The mean of the probability distribution $P(X = x)$ is denoted by μ, the variance is denoted by σ^2 and the standard deviation by σ.

Some of the most common probability distributions and their properties are listed below.

2.3.1.1 The Kronecker delta

If the random variable X can only take one possible value, i.e. if all nodes in a network have equal degree

$$P(X = x) = \delta(x, j) = \begin{cases} 1 & \text{if} \quad x = j \\ 0 & \text{otherwise.} \end{cases} \tag{2.6}$$

2.3.1.2 The uniform distribution

If every value in the range of the distribution is equally likely to occur, X follows the uniform distribution [38]. The uniform distribution for the range $[a, b]$ is given by:

$$P(X = x) = \frac{H(x - a) - H(x - b)}{b - a}, \tag{2.7}$$

where $H(x)$ is the Heaviside step function.

2.3.1.3 The binomial distribution

Definition 2.23. A *Bernoulli trial* is a random variable X with two possible outcomes, success or failure, and is associated with a success probability p.

The probability of obtaining n successes in N independent Bernoulli trials, where each trial X_i has success probability p, is given by the binomial distribution:

$$P(X_1 + \cdots + X_N = n) = \begin{pmatrix} N \\ n \end{pmatrix} p^n (1 - p)^{N-n}. \tag{2.8}$$

2.3.1.4 The normal distribution

The normal distribution is one of the most common probability distributions that models a wide range of phenomena. The distribution function has the shape of a bell. The normal distribution is given by:

$$P(X = x) = \frac{1}{\sigma\sqrt{2\pi}} e^{\frac{-(x-\mu)^2}{2\sigma^2}}. \tag{2.9}$$

2.3.1.5 The Poisson distribution

The Poisson distribution is generally used to model infrequent events. The Poisson distribution is given by:

$$P(X = x) = \frac{\mu^x e^{-\mu}}{x!}. \tag{2.10}$$

2.3.1.6 The exponential distribution

The exponential distribution has many applications such as modelling the decay in radioactivity. The exponential distribution is given by:

$$P(X = x) = \frac{1}{\mu} e^{-\frac{x}{\mu}}. \tag{2.11}$$

2.3.1.7 The power law distribution

The degree sequences of many real world networks follow a power law distribution [86]. In other words, many nodes in the network have a small degree, whereas very few nodes have a high degree. The power law distribution is given by:

$$P(X = x) = Cx^{-\alpha}, \tag{2.12}$$

where $\alpha > 0$ and C are constants. C normalises the distribution. For most real world networks $2 \geq \alpha \geq 3$ [86].

2.3.2 Generating functions

Generating functions are useful for the representation of probability distributions. A generating function is a representation of a sequence of numbers as a series in powers of a formal variable.

Definition 2.24. Let $p_0, p_1, p_2, \ldots$ be an arbitrary (infinite) sequence of numbers. The *generating function* for this sequence is the expression

$$f(x) \;=\; \sum_j p_j x^j.$$

If p_j is a probability function, for instance p_j could represent the probability of a node having degree j, $f(x)$ is called the probability generating function [86].

If the probability function p_j is normalised to unity, such that $\sum\limits_j p_j = 1$, then $f(1) = 1$.

Generating functions can be differentiated.

Definition 2.25. The *derivative* $\frac{d}{dx}f(x)$ of the generating function $f(x)$ is given by

$$\frac{d}{dx}f(x) = p_1 + 2p_2x + 3p_3x^2 + \cdots = \sum_j jp_jx^{j-1}$$

Letting $x = 1$, gives

$$\left[\frac{d}{dx}f(x)\right]_{x=1} = \langle j \rangle, \tag{2.13}$$

where $\langle j \rangle$ is the mean of the probability distribution or the average node degree in a network. Similarly, this result holds for higher moments:

$$\left[\left(x\frac{d}{dx}\right)^n f(x)\right]_{x=1} = \langle j^n \rangle. \tag{2.14}$$

We will frequently make use of Equations (2.13) and (2.14) in Chapters 3 and 4.

2.4 Datasets

This section provides some background on the datasets that we analyse in this thesis. More information on the data and previous studies regarding the datasets are provided in the individual chapters if needed. All datasets are publicly available apart from one (see Subsection 2.4.3) that we collected ourselves.

2.4.1 The 108$^{\text{th}}$ United States Senate network

Fowler [40, 41] studied the social connections between legislators in the United States of America by collecting data from the Library of Congress Thomas legislative database. He has made the data publicly available on his personal website (`http://jhfowler.ucsd.edu/cosponsorship.htm`). While co-sponsorship data of the U.S. Senate and U.S. House of Representatives for the 93$^{\text{rd}}$ to 108$^{\text{th}}$ Congresses can be accessed, we use the information available on the 108$^{\text{th}}$ U.S. Senate (January 2003 - January 2005).

	108$^{\text{th}}$ U.S. Senate network
Order, $\|U\| + \|V\|$	100 senators and 7,804 bills
Size, $\|E\|$	36,264
Density	0.0567
$\langle j \rangle$	362.64
$\langle k \rangle$	4.65

Table 2.1: General information about the 108$^{\text{th}}$ U.S. Senate network.

The U.S. Senate together with the House of Representatives constitutes the U.S. Congress. A total of 100 senators, two from every state, are voted into the U.S. Senate, where they can introduce a piece of legislation, called a bill. These bills can be co-sponsored by other members of the Senate. The U.S. Senate dataset may be represented as a bipartite network with 100 primary nodes, the senators, and 7,804 secondary nodes, the bills. An edge indicates that a senator has sponsored or co-sponsored a bill. Table 2.1 provides basic information about the dataset.

2.4.2 The Digg network

The Digg network [49] can be obtained at `http://konect.uni-koblenz.de/networks/digg-votes` and contains 3,018,197 votes by 139,409 users of the Digg website (`http://digg.com/`). The Digg website features news stories, with the option for users to vote for them. If a user votes for a story, it is understood that this user is interested in its contents. A total of 3,553 stories were rated over a period of one month in 2009. Table 2.2 contains basic information about the Digg dataset.

	Digg network
Order, $\lvert U \rvert + \lvert V \rvert$	139,409 users and 3,553 stories
Size, $\lvert E \rvert$	3,018,197
Density	0.0061
$\langle j \rangle$	21.65
$\langle k \rangle$	849.48

Table 2.2: General information about the Digg dataset.

2.4.3 The Facebook election data

The Facebook Graph API Explorer allows Facebook users to extract data from their website and can be accessed with the R package Rfacebook [7].

We extracted data of posts by 669 candidates of the 2016 Australian Federal election between 9$^{\text{th}}$ May 2016 and 2$^{\text{nd}}$ July 2016 (the period of election campaign) and constructed a bipartite network of Facebook users and politicians. A user is connected to a politician if the user liked at least one of the politicians' posts that was published during the election campaign.

	Facebook network
Order, $\lvert U \rvert + \lvert V \rvert$	669 candidates and 682,022 users
Size, $\lvert E \rvert$	1,311,206
Density	0.003
$\langle j \rangle$	1959.95
$\langle k \rangle$	1.99

Table 2.3: General information about the Facebook dataset.

For the 2016 Federal election a total of 1,648 candidates contested for 226 seats. The 669 candidates included in our dataset consists of all sitting members of the parliament (prior to the election) who had an active Facebook page. We also included all other candidates who were contesting marginal seats and had an active Facebook page. Marginal seats are electorates where the outcome of the poll is highly uncertain. The rest of the candidates were chosen randomly. Table 2.3 contains basic information about the Facebook dataset.

2.4.4 MovieLens

GroupLens (`http://grouplens.org/`) is a research lab at the University of Minnesota, collecting data from their MovieLens website (`http://movielens.org`) and making it publicly accessible. The MovieLens website allows users to rate and review movies. The website then recommends movies that the user may also be interested in.

2.4.4.1 The MovieLens 10M network

The MovieLens 10M dataset [47] contains 10,000,054 ratings ranging between 1 and 5, with 5 being the best possible rating. Starting in January 1995, 71,567 different users rated 10,681 movies over a period of 14 years. Every user has a unique id but no additional information about the users is known. All users that are included in the dataset rated at least 20 movies. Table 2.4 contains basic information about the MovieLens 10M dataset.

	MovieLens 10M network
Order, $\lvert U\rvert + \lvert V\rvert$	71,567 users and 10,681 movies
Size, $\lvert E\rvert$	10,000,054
Density	0.0131
$\langle j\rangle$	143.11
$\langle k\rangle$	936.60

Table 2.4: General information about the MovieLens 10M dataset.

2.4.4.2 The MovieLens tag genome data

The MovieLens tag genome dataset [125] contains tag relevance scores for 9,734 movies. There are 1,128 different tags and the relevance score ranges between zero and one, with one indicating strong relevance. Tags are words assigned to movies by users of the MovieLens website. Users can tag a movie with any word that they feel best describes the movie. The network is formed by connecting tags to movies. Edge weights record the relevance score of a tag to a movie.

	relevance score ≥ 0.5	100 most popular tags, relevance score ≥ 0.5
Order, $\lvert U\rvert + \lvert V\rvert$	1,128 tags and 9,734 movies	100 tags and 9,550 movies
Size, $\lvert E\rvert$	456,208	71,763
Density	0.0415	0.0751
$\langle j\rangle$	404.44	717.63
$\langle k\rangle$	46.87	7.51

Table 2.5: General information about the MovieLens tag genome dataset.

We consider two different networks. The first contains edges with weights greater or equal to 0.5. The second contains the 100 most popular tags, as determined by GroupLens, and edges with weights greater or equal to 0.5. Table 2.5 contains basic information about the two MovieLens tag networks.

2.4.5 The New South Wales crime network

The New South Wales Bureau of Crime Statistics and Research has made crime data collected in the state of New South Wales, Australia, publicly available (`http://data.gov.au/dataset/nsw-crime-data/`). The dataset records occurrences of crime in New South Wales between the years 1995 and 2012. When a crime occurs, its offence category, the month and local government area of occurrence is recorded. The New South Wales Bureau of Crime Statistics and Research provides a helpful visualisation tool for the dataset on their website (see `http://crimetool.bocsar.nsw.gov.au/bocsar/`).

	NSW crime network
Order, $\|U\|+\|V\|$	62 offence categories and 155 local government areas
Size, $\|E\|$	4,108
Density	0.4409
$\langle j\rangle$	67.77
$\langle k\rangle$	26.73

Table 2.6: General information about the NSW crime dataset. The values are average values over all 216 months.

We constructed 216 bipartite networks, one for every month between January 1995 and December 2012, of offence categories and local government areas from the data. New South Wales is divided into 155 local government areas and the data contains 62 different offence categories. Table 2.6 contains basic information about the NSW crime network. The values displayed in the table are average values over all 216 months.

2.4.6 The Noordin Top network

The Noordin Top data contains information about members of the Indonesian terrorist ring responsible for the 2003 JW Marriott hotel bombing in Jakarta, the 2004 Australian embassy bombing in Jakarta, the 2005 Bali bombings and the 2009 JW Marriott-Ritz-Carlton bombings [107]. We look at a sub-network of 26 members who attended 20 different meetings. The data is publicly available at `https://sites.google.com/site/sfeverton18/research/appendix-1` [35]. Table 2.7 contains basic information about the Noordin Top network.

	Noordin Top network
Order, $\lvert U\rvert + \lvert V\rvert$	26 members and 20 meetings
Size, $\lvert E\rvert$	64
Density	0.1231
$\langle j\rangle$	2.46
$\langle k\rangle$	3.20

TABLE 2.7: General information about the Noordin Top dataset.

2.4.7 The Southern Women network

The Southern Women network is perhaps the most famous and well studied bipartite network in the literature. The Southern Women dataset was collected by Davis et al. [27] to perform a study of social interactions among several women. The dataset contains information about 18 women attending 14 different social events. An edge between a woman and an event indicates that the woman attended the event. Table 2.8 contains basic information about the Southern Women network.

	Southern Women network
Order, $\lvert U\rvert + \lvert V\rvert$	18 women and 14 events
Size, $\lvert E\rvert$	89
Density	0.3533
$\langle j\rangle$	4.94
$\langle k\rangle$	6.36

TABLE 2.8: General information about the Southern Women network.

2.4.8 The United Kingdom crime network

In the United Kingdom crime data collected by the British police is publicly available online (`https://data.police.uk/`). Crimes are split into different categories. When a crime occurs, its category is recorded together with its location in form of latitude and longitude coordinates and the month of occurrence. The location of a crime is anonymised prior to entering the database in the following manner:

The police keeps 750,000 so-called map points, public locations such as bars, airports, shopping centres or the centre of a street. When a crime is recorded, it is matched to the nearest map point, thus anonymising its exact location. Tompson et al. [122] studied the spatial accuracy of the data and found that spatial errors are sizeable only if the data is studied at a small scale, such as at postcode level.

	January	February	March	April	May	June
Order, $\lvert U\rvert + \lvert V\rvert$	$5,944 + 3,037$	$5,553 + 3,019$	$5,473 + 2933$	$5,033 + 2,767$	$5,163 + 2,842$	$5,083 + 2,778$
Size, $\lvert E\rvert$	7,045	6,710	6,511	5,985	6,131	6,059
Density	0.0004	0.0004	0.0004	0.0004	0.0004	0.0004
$\langle j\rangle$	1.19	1.21	1.19	1.19	1.19	1.19
$\langle k\rangle$	2.32	2.22	2.22	2.16	2.16	2.18

TABLE 2.9: General information about the United Kingdom crime network, where U is the set of crimes and V is the set of locations.

We consider a subset of the data covering all burglaries in the area of Greater London (approximately 200 postcodes) during the period between January 2016 and June 2016 inclusively. Table 2.9 contains basic information about the data for each month.

Chapter 3

Significant Connections in One-mode Projections

Part of this chapter has been published in [69].

3.1 Introduction

3.1.1 Motivation

Most network measures cannot be applied directly to bipartite networks. Hence the most common path to analysing bipartite data is the examination of the projection onto one of its node sets. One way of constructing the one-mode projection is to drop one node set of the bipartite network and to connect two nodes of the remaining set if they share at least one neighbour in the observed bipartite network [90]. The process of projecting a bipartite network causes an inflation of edges. Since not all of these edges have significant meaning noise is added to the projection. Consequently, the results arising from the study of one-mode projections may be misleading.

One-mode projections are used in the study of recommendation systems to determine how similar any two, for example users of a rating system, are. If a user is similar to another, the recommendation system can recommend the items that one user liked to the other. One-mode projections also find applications in the search of new uses for pharmaceutical drugs. Drug-target networks are bipartite networks consisting of a set of drugs and a set of targets. Projection onto the set of drugs is used to find new targets for known drugs. However, the inflation of edges in the projection causes several problems in the identification of new drug applications [126].

3.1.2 Outline

Our contributions in this chapter are the following: We demonstrate that the edge weights of a projected bipartite network onto either one of its two node sets follow a Poisson binomial distribution. We then use this result to propose a novel technique to extracting the significant edges of one-mode projections. We further demonstrate that eliminating insignificant connections from a one-mode projection reveals its community structure.

Note that the aim of this chapter is not the partitioning of bipartite networks. The aim is to speed up the extraction of the backbone of one-mode projections which reveals the community structure of the projection. As the communities in the projection are partitions of a one-mode network we do not consider algorithms designed for bipartite networks.

The chapter is structured as follows: We start by critically reviewing different approaches to projecting bipartite networks in Section 3.2, to identify the key issues that arise. In Section 3.3 we propose a novel technique to eliminating the insignificant connections in one-mode projections. Our technique, based on the edge weight distribution of projections, is much faster than previous methods. In Section 3.4 we demonstrate our approach on several real world networks and observe that deleting insignificant edges of one-mode projections leads to a clearly visible community structure. This observation is discussed in Section 3.5. We conclude the chapter with a summary in Section 3.6.

3.2 One-mode projections and their limitations

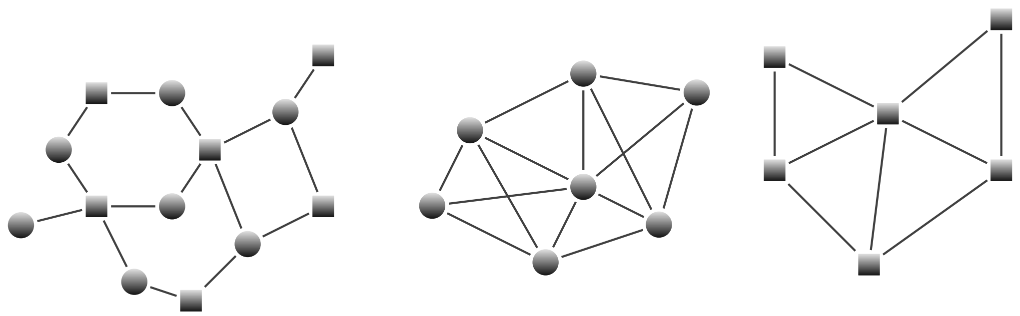
There are many ways of projecting a bipartite network onto one of its node sets. This section reviews the most popular methods and points out pitfalls and limitations.

3.2.1 The binary one-mode projection

The binary one-mode projection is the simplest and most straightforward means of projecting a bipartite network and is formally defined as follows:

Definition 3.1. Let $\mathcal{B}(U, V, E)$ be a bipartite network with the two disjoint node sets U and V and the edge set $E \subseteq U \times V$. The *one-mode projection* of $\mathcal{B}$ onto the node set U is the network $\mathcal{G}(U, E')$ with node set U and edge set E' such that $E' = \{(u, u') : \exists\ v \in V \text{ and } (u, v), (u', v) \in E\}$.

Note that every bipartite network has two projections (see Figure 3.1), one onto the primary node set U, called the primary projection, and one onto the secondary node set V, called the secondary projection.



(A) (B) (C)

FIGURE 3.1: A bipartite network (A) with its primary projection (B) and its secondary projection (C).

Despite the knowledge that one-mode projections lead to information loss [61, 126, 137, 11], their study is often preferred [37, 89, 109, 129] and frequently justified by the interest in only one of the two node sets. Watts and Strogatz [129], for instance, analyse a network of actors, where two actors are connected if they appeared together in at least one movie. Similarly, Ferrer and Solé [37] examine the properties of a network of words with connections inferred from their co-occurrence in sentences. By projecting onto the node set of interest, information encoded in the other set is usually disregarded. However, some researchers believe that even when being solely interested in one of the two node sets, the second set should not be ignored. Breiger [15] advises that one must consider the interplay between the primary and secondary node sets, since the secondary set carries valuable information about the primary set and vice versa. For example, a scientific author may be characterised by the papers he publishes. Thus, discarding the set of papers results in the deletion of the information about the papers as well as some of the information about their authors.

The perhaps most apparent limitation of the one-mode projection is that projections are not unique. The function that maps the set of all bipartite networks onto their, say primary projection, is not injective, meaning that many bipartite networks share the same primary projection [45]. The same holds for the secondary projection. Consider the two sub-graphs depicted in Figure 3.2A. In both cases, projection onto the primary node set results in the sub-graph displayed in Figure 3.2B. Hence, when presented with a network that is known to be a projection, in the absence of the original network it is impossible to deduce how the connections in the projection were formed. If both the primary and the secondary projection are available, it is in some cases possible to recover the original bipartite network. The following subsection discusses the details.

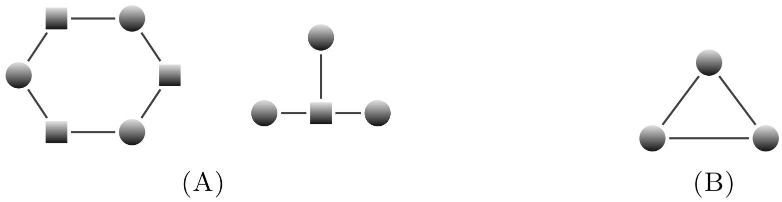


FIGURE 3.2: The primary one-mode projections (B) of the two sub-graphs shown in (A) are identical.

3.2.1.1 The dual projection approach

Everett and Borgatti [33] note that the original bipartite network may be recovered if both weighted projections are available. There are many possibilities to assign weights to a one-mode projection, some of which are outlined in Subsection 3.2.2. In [33] each edge connecting two nodes in the one-mode projection is associated with a weight that is equal to the number of the nodes' common neighbours in the bipartite network.

The reconstruction of a bipartite network from both its weighted projections is only possible in certain cases. Let $\phi : \mathbb{B} \to (\mathbb{G}, \mathbb{G})$ be the function that maps the set of all bipartite networks onto the set of combinations of their primary and secondary projections. Reconstruction of the original bipartite network is only possible if the combination of primary and secondary projections has exactly one pre-image. In other words, if there exist two non-isomorphic bipartite graphs $\mathcal{B}_1 \in \mathbb{B}$ and $\mathcal{B}_2 \in \mathbb{B}$ that have the same primary projection and the same secondary projection, recovery of $\mathcal{B}_1$ and/or $\mathcal{B}_2$ is impossible.

In fact, two non-isomorphic matrices with identical primary and secondary projections may be constructed as follows [33]:

Let D and P be two matrices of the same size, where D is any diagonal matrix with positive entries and P is any orthogonal matrix. Then, the matrices $B_1 = PDP^T$ and $B_2 = P(-D)P^T$ are non-isomorphic, but have the same primary and secondary projections. We wish to remark here that $B_1 = -B_2$. Although negative connections may represent for instance enmity, the case that all edges of a network are negative is very rare. It would be worth exploring if there are other ways of finding non-isomorphic matrices with identical primary and secondary projections and is left for future work.

As this particular way of construction results in non-binary bipartite networks, Everett and Borgatti [33] conclude that if the original bipartite network is binary, reconstruction is possible in most cases. One way of recovering a binary bipartite network is singular value decomposition. The $m \times n$ biadjacency matrix B of rank r that represents a given bipartite network may be written as:

$$B = USV^T, \tag{3.1}$$

where $U = [\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_m]$, $V = [\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_n]$ with $\mathbf{u}_i$ an eigenvector of BB^T for $1 \leq i \leq m$ and $\mathbf{v}_j$ an eigenvector of B^TB for $1 \leq j \leq n$. $S = \begin{bmatrix} D & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix}$, where D is an $r \times r$ diagonal matrix with the singular values of B along its diagonal. Everett and Borgatti [33] show that given the two projections of B, given by BB^T and B^TB (see Subsection 3.2.2), the original binary bipartite network may be recovered if the eigenvalues of BB^T are unique, by the below procedure:

1. Find the eigenvalues of BB^T and check that they are unique.
2. Calculate the unit length eigenvectors of BB^T and B^TB.
3. Construct matrix $B' = USV^T$. If B' is binary, then $B' \simeq B$ with high probability. If B' is not binary, change the signs of eigenvector $\mathbf{u}_i$ and repeat steps 1 and 2, for $i = 1, 2 \ldots m$ until B' is binary. Else change the sign of any two of the eigenvectors and so on. There are 2^{m-1} possible combination, one of which will result in a binary matrix B'.

This process is cumbersome and, as the authors of [33] themselves point out, the two projections are usually constructed from the original biadjacency matrix and hence there is no need to reconstruct it. The authors further accept that the results obtained from analysing both projections do not necessarily need to match the results obtained from an analysis of the bipartite network. On the other hand, being able to recover the original bipartite network from both its projections emphasises that much information is encoded in the interplay of primary and secondary nodes. In addition, there are further limitations that make the analysis of either one-mode projection problematic. These limitations are discussed next.

3.2.1.2 Concentration of cliques

In comparison to ordinary random one-mode networks, projections display a higher concentration of cliques [94]. By projecting a bipartite network onto its primary node set U, every node $v \in V$ of degree k induces $k(k-1)/2$ edges in the projection [61]. As a direct consequence the concentration of cliques (see Definition 2.7) in the projection as well as the projection's density are much higher than expected in a random network. In fact, the projection of a star-sub-graph (see Definition 2.8) of any order is a clique, since all of the star-sub-graph's primary nodes are connected to the same secondary node (see Figure 3.3).

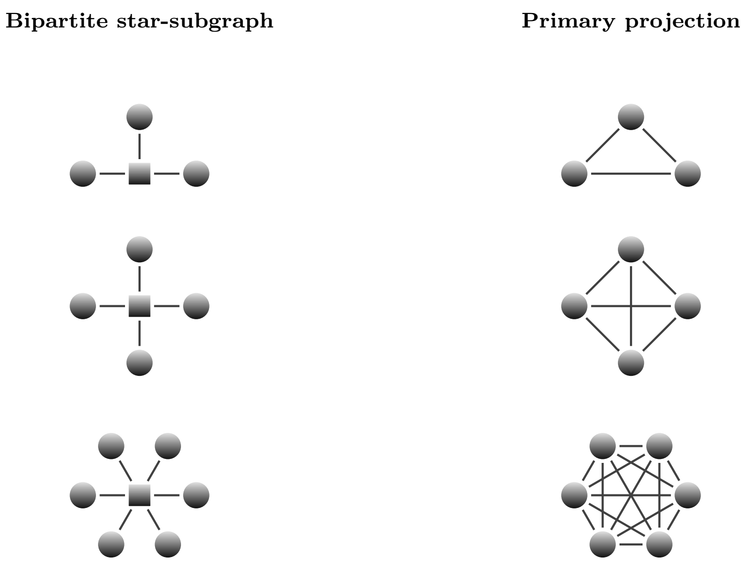


FIGURE 3.3: The projection of a star sub-graph of any order results in a clique.

The high concentration of cliques has great impact on measures such as the clustering coefficient. The clustering coefficient measures the concentration of triangles in a given network, and hence, is higher in a projected network than one would expect to observe in a similar random network. This was demonstrated by Opsahl [94], by showing that the concentration of triangles in a projected collaboration network is 350 times higher than expected. Indeed, many early studies of one-mode projections claim a higher than expected clustering coefficient overlooking the fact that the concentration of triangles is naturally much higher in projected bipartite networks [16, 56, 63, 105]. This particular problem is dealt with in Chapter 4.

The search for significant patterns that require comparison to random networks is also affected by the high concentration of cliques. A particular sub-graph that occurs with a higher probability in a given network than expected in a random network of similar type is called a motif [115]. As one would detect many more cliques in a projection than in a random one-mode network, the occurrence of motifs is strongly biased in projected networks. A possible way of avoiding this bias is by randomising the observed bipartite network and then projecting, instead of randomising the projection (see Subsection 4.3.2). With a single one-mode projection running in $\mathcal{O}(|U|^2|V|)$ time, this approach becomes quickly infeasible as the network grows. Note that there exist algorithms for fast matrix multiplication [132] and although the computation time is slightly lowered one-mode projections remain impractical as the order of the network grows.

3.2.1.3 Density of one-mode projections

A related problem caused by the projection of bipartite networks is an inflation in the number of edges. Projections are often not only dense, but noisy networks. We consider a network as noisy if relevant information is hidden. Since the edges in a one-mode projection are indirectly created, many of them may not have a significant meaning and will thus hide the important connections. As an example, consider a network of people and events. When projecting onto the set of people, each event creates $k(k-1)/2$ links amongst its k attendees, making the the projection dense (see Figure 3.3, with circles representing people and squares representing events). As the act of projection builds these connections indirectly, two people who attend the same event need not necessarily have a connection in reality. This is especially the case for large events with many participants. Similarly, if the network is projected onto the set of events, every person creates links between the events he attended, again producing a dense one-mode projection. For example, a person with a number of different interests would create connections amongst many different events. Noise is thus added to the network as the events may not have a lot in common.

The high density and noisiness of one-mode projections present several challenges. Since the edges are indirectly inferred, the most relevant information may be hidden and hard to recognise. Before tackling the problem of revealing the true structure of one-mode projections, we review methods that result in more informative one-mode projections in the following subsection. Without loss of generality, only primary projections are discussed.

3.2.2 Weighted projections

Despite the many problems that arise when projecting bipartite networks, researchers seem to prefer the analysis of one-mode projections and tend to tackle some of the issues by associating edges in the projection with weights to create a more informative network. The question of determining the edge weights in the projection immediately arises [137].

The simplest method of assigning weights is to associate an edge connecting two nodes in the one-mode projection with a weight that is equal to the number of previously shared neighbours [128]. Unless otherwise specified, the term *weighted one-mode projection* refers to this type of projection.

The weighted one-mode projection is obtained by multiplying the network's biadjacency matrix B with its transpose B^T. Thus, the weighted one-mode projection onto the primary node set is given by BB^T and the weighted projection onto the secondary node set is given by B^TB. Although this type of projection is more informative than the binary one-mode projection, it still has drastic limitations [91]. For instance, two scientists who are the authors of a paper that has many other co-authors may not have a very strong relationship compared to two scientists who are the only two co-authors of an article. A weighted projection would give the same weight to such different relationships. This has resulted in various weighting methods being proposed, many of which have been discussed in the context of collaboration networks.

3.2.2.1 Newman's approach

Newman [91] proposes a weighting method that predicts the strength of connections in projected collaboration networks. He makes the following assumptions: Two scientists who are the sole authors of a paper know each other better than scientists who are part of a larger collaboration. In addition, two authors who have collaborated many times have a stronger connection than two authors who wrote only a few papers together.

According to Newman's [91] assumptions, an author shares his time equally amongst his co-authors. Hence, he proposes a weighting method that inversely associates an edge with a weight according to the total number of co-authors. Mathematically, the weight $\omega_{uu'}$ of the edge that connects authors u and u' in the one-mode projection is given by:

$$\omega_{uu'} = \sum_{v} \frac{b_{uv} b_{u'v}}{\deg(v) - 1}, \tag{3.2}$$

where $b_{uv} = 1$ if author u has co-authored paper v and 0 otherwise. Papers with a sole author do not contribute to the collaboration network and are hence disregarded. Note that the diagonal entries of the projection are set to 0.

Newman [91] is particularly interested in the closeness centrality (see Equation (2.4)) of scientists in collaboration networks. The closeness centrality score of a node is the inverse of its average distance to all other nodes in the network. Newman [91] compares the closeness scores of scientists in simple binary projections to the closeness scores of scientists in weighted projections, where weights are assigned to the edges according to Equation (3.2). His results show that, in simple binary projections, as the number of different co-authors of a scientist grows, his average distance to other scientists becomes smaller. In weighted projections on the other hand, having a large number of collaborators no longer implies a short average distance to all other scientists in the network. In order to achieve a short average distance it is more important to have strong connections to co-authors that are themselves well connected.

3.2.2.2 Li et al.'s approach

Li et al. [64] choose a different approach to achieve the weighting of the one-mode projection of a network. Like Newman [91] they assume that two authors who have written a large number of papers together, know each other very well. They further assume that the contribution of each additional paper to the strength of the connection between the two authors, diminishes. In other words, Li et al. [64] assume a saturation effect with the increase in the number of collaborations and thus use the hyperbolic tangent function $\tanh(x)$ to describe this effect. Hence, the edge that connects authors u and u' has weight

$$\omega_{uu'} = \tanh(B[u,] \cdot B[u',]), \tag{3.3}$$

where $B[u,]$ is the row of the biadjacency matrix B that corresponds to node u and hence $B[u,] \cdot B[u',]$ is the number of papers that author u has co-authored with author u'. Note that the authors could have chosen other functions of similar shape instead of the hyperbolic tangent.

3.2.2.3 Zhou et al.'s approach

Another way of making the one-mode projection of a bipartite network more informative is given in [137]. Zhou et al. [137] introduce an asymmetrical projection that gives higher importance to publications with a single author, instead of discarding them, as done by [91]. Edges are associated with two different weights $\omega_{uu'}$ and $\omega_{u'u}$, where $\omega_{uu'} \neq \omega_{u'u}$. This particular manner of weighting the edges in the projection comes naturally, as two co-authors may feel differently about their strength of collaboration. In contrast to [91], Zhou et al. [137] believe that inclusion of single authored papers makes the projection more informative and is important, considering that over 50% of mathematical review papers have a sole author [44].

To construct the weighted primary projection Zhou et al. [137] assume that a certain amount of resource is allocated to each primary node. A node's resource is equally shared and sent to its neighbours and hence, each secondary node has some amount of resource available that is again equally shared and sent back to their neighbours (see Figure 3.4).

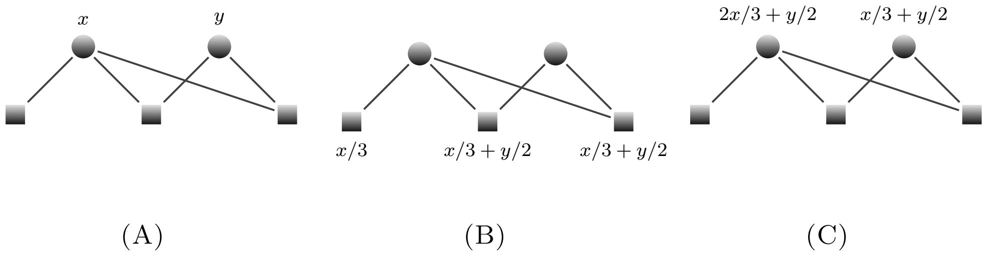


FIGURE 3.4: Each node is given a certain amount of resource. Here, the two primary nodes have resources x and y respectively (A). Each primary node shares its available resource equally amongst its neighbours. For example, the node having resource amount x, sends amounts $x/3$ to each of its three neighbours (B). In the final step of the resource allocation process, the secondary nodes share their resources equally amongst their neighbours (C) [137, p.3]

In this manner each edge in the projection is assigned two weights (see Figure 3.5). Mathematically, the weight $\omega_{uu'}$ that node u' assigns to node u is given by

$$\omega_{uu'} = \frac{1}{\deg(u')} \sum_v \frac{b_{uv} b_{u'v}}{\deg(v)}, \tag{3.4}$$

where $b_{uv} = 1$ if node u and node v are connected in the bipartite network and 0 otherwise.

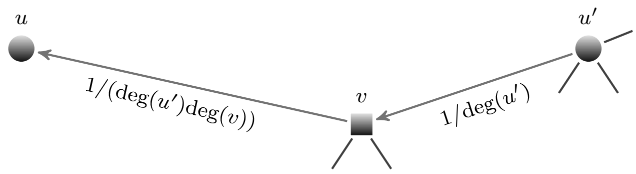


FIGURE 3.5: Node u' sends resource amount $1/\deg(u')$ to node v. Since node v equally shares its resource amongst its neighbours, $1/(\deg(u')\deg(v))$ of node u's resource reaches node u, given that v is the only common neighbour of nodes u and u'. The edge pointing from node u' to node u in the weighted projection thus has weight $1/(\deg(u')\deg(v))$.

Note that the diagonal elements in the projection matrix are non-zero and contain the importance that an author assigns to himself. In this manner, single authored papers are incorporated into the projection.

Zhou et al.'s [137] approach of weighting a network has been developed for a particular application, that of recommendation systems. They propose a personal recommendation algorithm, based on their weighted one-mode projection, that performs better than previously proposed algorithms such as collaborative filtering [48] and the global ranking method.

Many other weighting methods have been suggested [eg. 9, 72], each providing interesting insights into networks with regards to different applications. While weighted projections are undoubtedly useful for some specific applications such as the design of recommendation algorithms [137], often, more sophisticated weighting methods result in networks that are harder to analyse. For example, the approach by Zhou et al. [137] results in networks with fractional weights. In addition, real weighted networks frequently display irregularly distributed edge weights. As a result, connections with a low weight can be highly important on a local scale [39]. It follows that weighted projections do not always capture the true underlying relationship between the entities of a system. However, there exist techniques to extract the most important connections of a network. These are presented in the following section.

3.3 Backbone extraction

One-mode projections are generally dense and often contain many edges that are redundant or insignificant, which is apparent from the often complete graph that results from one-mode projections. Thus the underlying structure of a projected network is very likely hidden. Identifying and then discarding insignificant edges would allow the true underlying structure to become apparent. This process is called backbone extraction. A well extracted backbone of a network would greatly simplify the analysis of one-mode projections as such a backbone would contain only the most relevant information. Firstly, we give a definition of the backbone:

Definition 3.2. The *backbone* of a network $\mathcal{G}(U, E)$, with node set U and edge set E, is defined as the sub-graph $\mathcal{G}'(U, E')$ of $\mathcal{G}$, such that the edge set E' of the backbone $\mathcal{G}'$ contains only the most significant edges in E.

The definition of the backbone clearly states that an edge needs to be significant for inclusion in the backbone. The identification of significant edges is a challenging task, since no one has, as yet, clearly defined what makes an edge significant.

Intuitively one may think that the most significant edges in a network are those associated with the highest weights. As Foti et al. [39] say, it is common practice to reduce noise in networks by dropping edges with weights under a certain threshold. As low edge weights may very well represent strong connections [82], these thresholds are thought of as naive thresholds. Consequently more sophisticated methods are needed to reduce the number of edges in dense networks.

The aim of this section is to provide a clear definition of edge significance. We further present a novel method of extracting the backbone of one-mode projections in Subsections 3.3.3 and 3.3.4 and demonstrate that edge weight and edge significance generally show a weak correlation in Section 3.4, thus pointing out the limitations of applying a global threshold to filter out redundant information.

3.3.1 The backbone of weighted one-mode projections

The idea of extracting the backbone of a given network was first discussed in terms of weighted one-mode networks that are not projections [39, 113, 114, 118, 136]. Serrano et al. [114] for example, emphasise that the edge weight probability distribution of a weighted network is often broadly distributed. The authors point out that the application of a naive threshold would hence filter out relevant structural information. They propose the determination of edge significance by comparison to a null-model. Foti et al. [39] use edge weight distributions instead of a null-model, leading to the identification of significant edges that build locally strong connections.

Comparable approaches have been suggested for weighted one-mode projections [43, 81, 82, 138]. We remind the reader that the term *weighted one-mode projection* refers to the weighted projection given by BB^T, where B is the biadjacency matrix of the network and B^T is its transpose.

Weighted one-mode networks that are projections of bipartite networks are very different from ordinary one-mode networks. The edge weights of one-mode projections directly depend on the node degrees of the bipartite network, thus constraining the range of weights that an edge in the projection can take [82]. Neal [82] introduces a model that takes the degrees of primary and secondary nodes into account to determine the significance of edges in the weighted one-mode projection. The range of possible weights of the edge connecting nodes u and u' in the weighted projection can be expressed as follows:

$$\max(0, j_u + j_{u'} - |V|) \leq \omega_{uu'} \leq \min(j_u, j_{u'}), \tag{3.5}$$

where j_u and $j_{u'}$ are the degrees of nodes u and u' respectively and $\omega_{uu'}$ is the weight associated with the edge connecting the two nodes in the weighted projection. We give a short proof of Inequality (3.5):

Proof. Let u and u' be two primary nodes of some bipartite network with degrees j_u and $j_{u'}$ respectively. The edge connecting u and u' in the weighted projection onto the primary node set has weight $\omega_{uu'}$ equal to the number of the common neighbours of u and u' in the bipartite network.

Since u has exactly j_u neighbours in the bipartite network and u' has exactly $j_{u'}$ neighbours, u and u' can have at most $\min(j_u, j_{u'})$ neighbours in common and hence $\omega_{uu'} \leq \min(j_u, j_{u'})$.

If $|V| \geq j_u + j_{u'}$, then $\omega_{uu'} \geq 0$. If $|V| < j_u + j_{u'}$, u and u' need to share at least $j_u + j_{u'} - |V|$ neighbours, by the pigeon hole principle [17]. Hence, $\omega_{uu'} \geq \max(0, j_u + j_{u'} - |V|)$. $\square$

To exemplify that connections with relatively low weight may be significant, consider two nodes u and u' with degrees $\deg(u) = 2$ and $\deg(u') = 3$ in a large bipartite network. Assume that the two nodes are connected in the projection via an edge with maximum possible weight, $\omega_{uu'} = 2$. Although the edge weight is relatively small, the connection would be considered significant as it is unlikely to be observed at random. The use of a naive threshold would remove such connections and hence it is important to consider Inequality (3.5) when determining the significance of an edge.

Neal [82] uses the following steps to extract the backbone of the weighted one-mode projection. First, the observed bipartite network is projected onto a weighted one-mode network. Second, a set of random bipartite networks is generated, each of which is projected onto a weighted one-mode network in the third step. Finally, the random projections are compared to the projection of the observed bipartite network. Any edge that displays a weight higher than expected is included in the backbone.

A stochastic degree sequence model (SDSM) is applied to generate the random bipartite networks. The SDSM aims to estimate the probabilities of any given primary node being connected to any given secondary node. A binary outcome model [80] is used to predict the entries of the biadjacency matrix. Binary outcome models find applications in the prediction of events that have two possible outcomes. Preferably, the model would closely estimate the degree distributions of the bipartite network. The fitted model is then used to find the probabilities π_{uv} of an edge occurring between a primary node u and a secondary node v in the bipartite network. A biadjecency matrix is then constructed with its entries being Bernoulli trials (see Definition 2.23) with success probability π_{uv}.

Neal [82] verifies his approach by extracting the backbone of the projection of the 108$^{\text{th}}$ U.S. Senate network [40, 41] (see Subsection 2.4.1 in Chapter 2 for a description of the dataset). Despite successfully identifying significant edges, it is computationally expensive; The one-mode projection of a bipartite network is obtained by multiplying its biadjacency matrix B with its transpose, which runs in $\mathcal{O}(|U|^2|V|)$ time. As Neal [82] opines, a method to directly calculate edge weight distributions would be highly beneficial. In the following section, we show that this is indeed possible by demonstrating that the edge weights in most random one-mode projections, onto either the primary or secondary node set, are distributed according to a Poisson binomial distribution. We develop a fast method of extracting the backbone of a one-mode projection without relying on the generation of random networks and their projections, thus reducing the computation time significantly. We corroborate the accuracy of the method by comparing the weights thus obtained to the real weight distribution of the projections of several types of randomly generated bipartite networks.

3.3.2 The Poisson binomial distribution

We begin with the necessary definitions and notation.

The probability of obtaining n successes in N independent Bernoulli trials (see Definition 2.23), where each trial X_i has success probability p, is given by the binomial distribution [38]:

$$P(X_1 + \cdots + X_N = n) = \binom{N}{n} p^n (1-p)^{N-n}. \tag{3.6}$$

If the N trials have varying probabilities p_i, where $i = 1, \ldots, N$, the sum of the independent, non-identically distributed random variables $X_1, \ldots, X_N$ is given by the Poisson binomial distribution [127]:

Let $\mathcal{S}_n$ be the set of all combinations of n distinct integers chosen from $\{1, \ldots, N\}$ and let $S_1, \ldots, S_{|\mathcal{S}_n|}$ be the elements of $\mathcal{S}_n$, where $|\mathcal{S}_n| = \begin{pmatrix} N \\ n \end{pmatrix}$. Let s denote an element of the subset S_j, where $1 \leq j \leq |\mathcal{S}_n|$ and let $\bar{S}_j$ denote the complement of S_j with respect to $\{1, \ldots, N\}$. Then the probability density function of the Poisson binomial random variable $Z_X = \sum\limits_{i=1}^{N} X_i$ is given by

$$P(Z_X = n) = \sum_{j=1}^{|\mathcal{S}_n|} \prod_{s \in S_j} p_s \prod_{\bar{s} \in \bar{S}_j} (1 - p_{\bar{s}}). \tag{3.7}$$

3.3.3 Approximation of the weight distribution

We now look at the use of the Poisson binomial distribution to approximate the distribution of weights in one-mode projections.

Let $\mathcal{B}$ be a bipartite network with the two disjoint node sets U and V. Let p_j denote the probability that a node $u \in U$ has degree j and let q_k denote the probability that a node $v \in V$ has degree k.

By Definition 2.24, $f(x) = \sum_{j=0}^{\infty} p_j x^j$ is the probability generating function of the primary node degrees and $g(x) = \sum_{k=0}^{\infty} q_k x^k$ is the probability generating function of the secondary node degrees. Generating functions are a useful tool to describe and calculate certain properties of complex networks. Their use has led to many interesting results. The most important properties of generating functions are listed in Subsection 2.3.2.

The average degree of the primary nodes, the first moment of $f(x)$, is denoted $\langle j \rangle$. Similarly, $\langle k \rangle$ denotes the average degree of the secondary nodes. In general, the n^{th} moment $\langle j^n \rangle$ of the degree distribution may be calculated as follows:

$$\left(x \frac{d}{dx} \right)^n f(x) \bigg|_{x=1} = \sum_{j=0}^{\infty} j^n p_j = \langle j^n \rangle. \tag{3.8}$$

Note that $\langle j^n \rangle \neq \langle j \rangle^n$.

The probability of an edge connecting a primary node to a secondary node in a bipartite network is determined by dividing the product of their degrees by the number of edges in the network. The number of edges m in a bipartite network is given by

$$m = |U|\langle j\rangle = |V|\langle k\rangle. \tag{3.9}$$

If $\pi_{uu'v}$ denotes the probability that two primary nodes u and u' are connected to a secondary node v, then given that $\deg(u) = j_u$, $\deg(u') = j_{u'}$ and $\deg(v) = k_v$,

$$\begin{aligned}\pi_{uu'v} &= \frac{j_u j_{u'} k_v(k_v-1)}{m(m-1)} \\ &= \frac{j_u j_{u'} k_v(k_v-1)}{|U|^2\langle j\rangle^2 - |U|\langle j\rangle}.\end{aligned} \tag{3.10}$$

Since p_j is the fraction of nodes with degree j in the primary node set, multiplying j_u and $j_{u'}$ by their respective probabilities and averaging over j_u and $j_{u'}$, results in the probability π_v that any two primary nodes are connected to a particular secondary node v of degree k_v. Hence,

$$\begin{aligned}\pi_v &= \sum_{j_u, j_{u'}} \frac{j_u p_{j_u} j_{u'} p_{j_{u'}} k_v(k_v-1)}{(|U|^2\langle j\rangle^2 - |U|\langle j\rangle)} \\ &= \frac{k_v(k_v-1)}{|U|^2\langle j\rangle^2 - |U|\langle j\rangle}\left[\sum_{j=0}^{\infty} j p_j\right]^2 \\ &= \frac{\langle j\rangle k_v(k_v-1)}{|U|^2\langle j\rangle - |U|}.\end{aligned} \tag{3.11}$$

The probability π_v is associated with the Bernoulli random variable X_v indicating the existence of a connection between two primary nodes via a particular secondary node v.

It follows that the probability of a randomly chosen edge in the one-mode projection having weight ω is given by

$$P\left(\Omega_X = \sum_{v=1}^{|V|} X_v = \omega\right) = \sum_{j=1}^{|\mathcal{S}_\omega|} \prod_{v \in S_j} \pi_v \prod_{\bar{v} \in \bar{S}_j} (1 - \pi_{\bar{v}}), \tag{3.12}$$

where $\mathcal{S}_\omega$ is the set of all combinations of ω integers chosen from $\{1, \ldots, |V|\}$.

As an example consider the network of two primary nodes u and u' and three secondary nodes. What is the probability of node u being connected to node u' by an edge of weight two? To answer this question, we need to find the probability of nodes u and u' sharing exactly two neighbours in the bipartite network. There are $\begin{pmatrix} 3 \\ 2 \end{pmatrix} = 3$ different possibilities (see Figure 3.6).

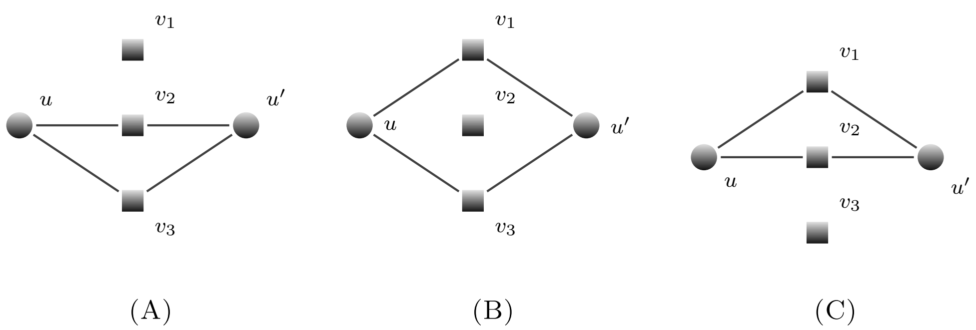


FIGURE 3.6: There are three different possibilities of getting an edge of weight two between nodes u and u' in a bipartite network with three secondary nodes.

Assuming that the probabilities π_v, that nodes u and u' are connected via node v, are equal for all $v = 1, 2, 3$, the probability of node u and u' sharing exactly two neighbours in the bipartite network is given by $P(\omega_{uu'} = 2) = \begin{pmatrix} 3 \\ 2 \end{pmatrix} \pi_v^2(1-\pi_v)$. Since the probabilities π_v are in general different, we have $P(\omega_{uu'} = 2) = \pi_1\pi_2(1-\pi_3) + \pi_1\pi_3(1-\pi_2) + \pi_2\pi_3(1-\pi_1)$, which can be generalised to Equation (3.12).

Since $P(\Omega_X = \omega)$ is hard to compute, we use the Poisson approximation instead:

$$P(\Omega_X = \omega) \approx \frac{\mu^\omega e^{-\mu}}{\omega!}, \tag{3.13}$$

where $\mu = \sum_{v=1}^{|V|} \pi_v$.

The error of the Poisson approximation of $P(\Omega_X = \omega)$ is given by $\epsilon_\omega < 2 \sum_{i=1}^{|V|} \pi_v^2$ and is small if the number of expected successes is small [62]. Since most real world networks are sparse, the Poisson approximation estimates the probability of weight ω very well.

The mean μ of the distribution is calculated as follows:

$$\begin{aligned} \mu &= \sum_{v=1}^{|V|} \pi_v \\ &= \frac{\langle j \rangle}{|U|^2 \langle j \rangle - |U|} \sum_{v=1}^{|V|} k_v (k_v - 1) \\ &= \frac{|V| \langle j \rangle \left(\langle k^2 \rangle - \langle k \rangle \right)}{|U|^2 \langle j \rangle - |U|}. \end{aligned} \tag{3.14}$$

3.3.4 Determining probabilities of individual connections

When extracting the backbone of a network, one is interested in the probability of observing a connection with a certain weight between two nodes u and u' in the projection, denoted by $P_{uu'}(\Omega_X = \omega)$.

If $\pi_{uu'v}$ is small for every $v = 1, \ldots, |V|$, the Poisson approximation may be used, with

$$\begin{aligned} \mu &= \sum_{v=1}^{|V|} \pi_{uu'v} \\ &= \sum_{v=1}^{|V|} \frac{j_u j_{u'} k_v (k_v - 1)}{|U|^2 \langle j \rangle^2 - |U| \langle j \rangle} \\ &= \frac{|V| j_u j_{u'} \left(\langle k^2 \rangle - \langle k \rangle \right)}{|U|^2 \langle j \rangle^2 - |U| \langle j \rangle}. \end{aligned} \tag{3.15}$$

In bipartite networks where some nodes have a very high degree, it is often found that the probability of a connection between two individual nodes is very high, resulting in large approximation errors. In such situations, instead of the Poisson approximation, we use the normal approximation:

$$P_{uu'}(\Omega_X = \omega) \approx \frac{1}{\sigma\sqrt{2\pi}} e^{-(\omega-\mu)^2/(2\sigma^2)}, \tag{3.16}$$

where μ is given by Equation (3.15) and

$$\begin{aligned}
\sigma &= \left[\sum_{v=1}^{|V|} \pi_{uu'v}(1-\pi_{uu'v})\right]^{1/2} \\
&= \left[\sum_{v=1}^{|V|} \pi_{uu'v} - \sum_{v=1}^{|V|} \pi_{uu'v}^2\right]^{1/2} \\
&= \left[\mu - \sum_{v=1}^{|V|} \left(\frac{j_u j_{u'} k_v(k_v-1)}{|U|^2\langle j\rangle^2 - |U|\langle j\rangle}\right)^2\right]^{1/2} \\
&= \left[\mu - \frac{|V| j_u^2 j_{u'}^2 \left(\langle k^4\rangle - 2\langle k^3\rangle + \langle k^2\rangle\right)}{|U|^4\langle j\rangle^4 - 2|U|^3\langle j\rangle^3 + |U|^2\langle j\rangle^2}\right]^{1/2}.
\end{aligned} \tag{3.17}$$

In order to demonstrate the accuracy of our approximation, we consider bipartite networks from all 25 possible permutations of the following degree distributions for U and V: The delta function, the uniform distribution, the normal distribution, the exponential distribution and the power law distribution. We project each permutation onto a one-mode network (by multiplying its biadjacency matrix B by its transpose) to determine their average weight distribution. To test our approximation for robustness, we vary the bipartite degree distribution and network parameters. For each variation we generated 100 random bipartite networks. Details on the generation of the random networks can be found in Chapter 2 (see Subsection 2.2.5.3).

We used the Kolmogorov-Smirnov (KS) test to compare the observed average weight distribution in the random networks to the approximated weight distribution. The KS test is a statistical test that allows the comparison of two distributions, with its null-hypothesis stating that the two sample distributions are drawn from the same distribution. The results of the KS tests suggest that the approximation estimates the expected weight distribution of a projected bipartite network extremely well and is robust against variation of the degree distribution and network parameters. As expected, our approximation performs poorly only for very dense bipartite networks, with p-values falling below 0.5. As most real world networks are extremely sparse such cases are rarely observed [57, 86].

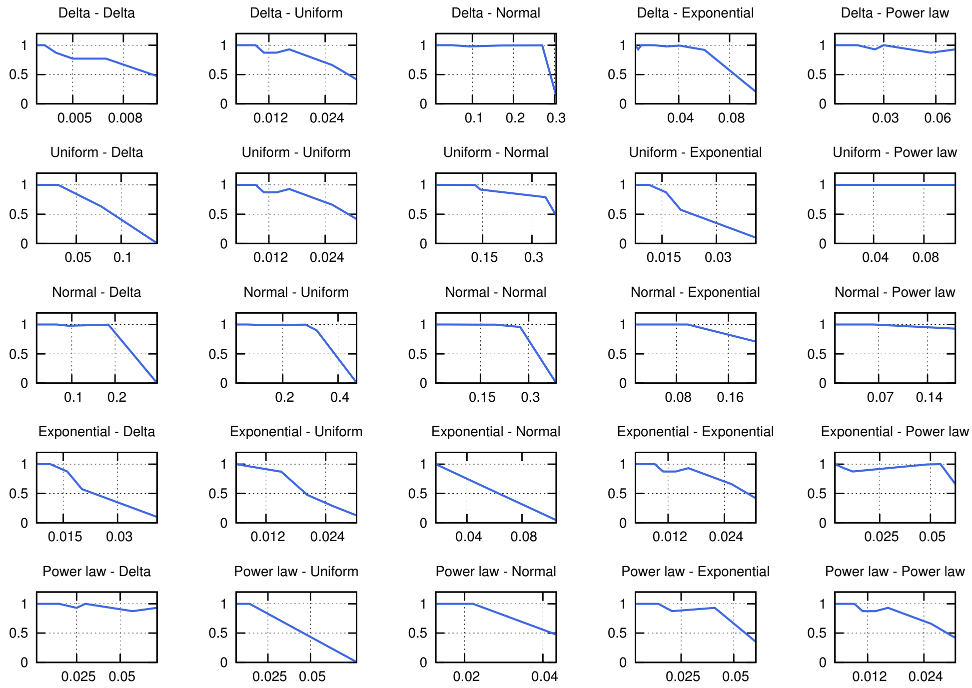


FIGURE 3.7: Results of the KS tests for 25 tested permutations of degree distributions. The x-axis displays the density and the y-axis the corresponding p-Value.

Figure 3.7 shows the results of the KS tests for the 25 tested permutations of degree distributions.

The weight probability distribution of an edge between two individual nodes u and u' is given by Equation (3.16). Once this distribution is calculated, the observed edge weight is compared to the distribution and regarded significant, in this research, if it is larger than the mean plus three standard deviations of the approximated distribution.

Definition 3.3. An edge in the weighted one-mode projection is *significant* if its weight is larger than the mean plus three standard deviations of the distribution given by Equation (3.16).

The threshold of three standard deviations is customary in many contexts and corresponds to a 95% confidence interval. However, one may wonder what happens with different thresholds. On assessing this sensitivity we found that lower thresholds retained many insignificant edges. Higher thresholds, on the other hand, removed a large number of edges, leading to the risk of deleting connections that were key to the network's topology. Thus, setting the threshold too high could result in the fragmentation of the network.

In summary, to extract the backbone of a one-mode projection, the weight distribution $P_{uu'}(\Omega_X = \omega)$ is computed for every edge in the network, resulting in a computation time of $\mathcal{O}(|U|^2)$. The greatest advantage of this method is that it avoids the necessity to generate any random networks, saving the time required to generate and then project hundreds or thousands of networks. Since a single projection runs in $\mathcal{O}(|U|^2|V|)$ time our approach greatly simplifies and speeds up the process of extracting the backbone of a one-mode projection.

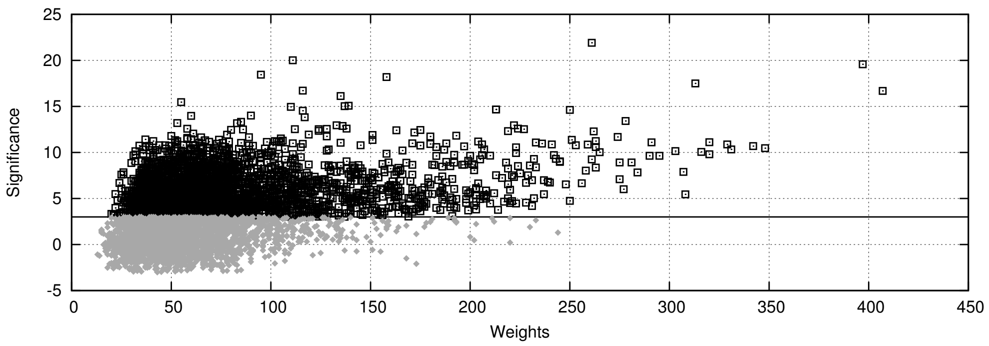
3.4 Backbone extraction of real world networks

In this section we extract the backbone of the projection of several real world networks: The 108$^{\text{th}}$ U.S. Senate network [40, 41], the MovieLens Tag Genome network [125] and a network of Facebook users and candidates of the 2016 Australian Federal election. Brief descriptions of these datasets are given in Chapter 2 (see Section 2.4). In the next section we will illustrate the usefulness of the backbone extraction in identifying communities within these datasets.

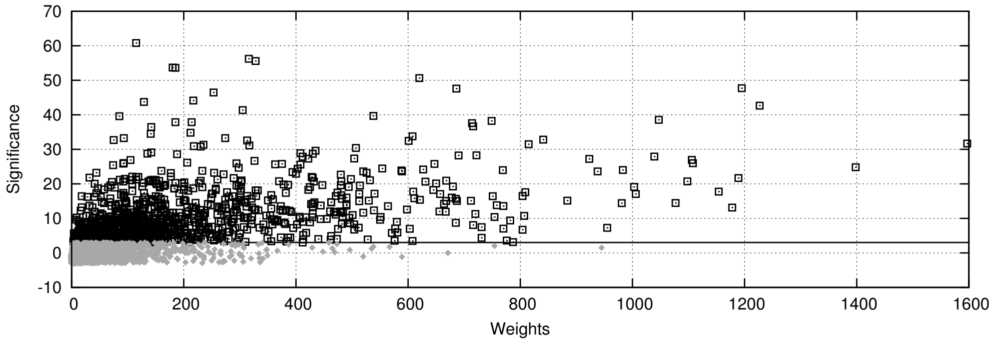
The U.S. Senate together with the House of Representatives constitutes the U.S. Congress. In every state two senators are voted into the senate, allowing them to introduce a piece of legislation, called a bill, that can be co-sponsored by other members of the senate. The U.S. Senate dataset may be represented as a bipartite network with 100 primary nodes, the senators, and 7,804 secondary nodes, the bills. An edge indicates that a senator has sponsored or co-sponsored a bill [40, 41]. This data set is publicly available and can be downloaded from `http://jhfowler.ucsd.edu/cosponsorship.htm`. For a more details of this dataset, refer to Chapter 2, Subsection 2.4.1. Here we consider the projection onto the set of senators.

The MovieLens Tag Genome dataset [125] was collected by the University of Minnesota and is available for download at `http://grouplens.org/datasets/movielens/tag-genome/`. This dataset contains 9,734 movies and 1,128 tags. Tags are words assigned to movies by users of the MovieLens website. Users may use as a tag any word that they feel best describes a movie. Edges connect tags to movies and record the strength of the association of a particular movie with a particular tag. Edge weights range between zero and one, where one indicates strong relevance. More details of this data can be found in Chapter 2 (see Subsection 2.4.4.2). Here, we consider the complete network as well as the subset of the 100 most popular tags (popularity as determined by members of the GroupLens research group, who also collected the data, `http://grouplens.org/`). Edges are only included if the tag relevance is greater or equal to 0.5. We consider the projection onto the set of tags.

The Facebook dataset contains posts from the Facebook pages of politicians who were candidates in the 2016 Australian Federal election. We constructed a bipartite network of Facebook users and politicians that were part of the 2016 Australian Federal Election. We collected data from the Facebook pages of 669 political candidates during the election campaign (9$^{\text{th}}$ May - 2$^{\text{nd}}$ July 2016) and linked a user to a politician if the user liked at least one of the politician's posts during the election campaign. The final network consists of 669 politicians, 682,022 Facebook users and 1,378,641 edges. A more detailed description of the dataset and its collection can be found in Chapter 2 (see Subsection 2.4.3).



(A)



(B)

FIGURE 3.8: Edge significances in the U.S. Senate network (A) and the MovieLens network (B) plotted against their corresponding edge weights. Significant edges are represented by black squares. The observed correlation is weak in both cases (0.3653 for the U.S. Senate network and 0.554 for the MovieLens network).

Extracting the backbone of the projections would reveal significant connections between pairs of nodes. Thus, in case of the U.S. Senate network one would expect to find the majority of significant links between senators of the same party. Similarly, the backbone of the tag-tag projection of the MovieLens dataset should reveal closely related tags.

	Projection	Backbone	Reduction in size
U.S. Senate	1	0.522	47.8%
MovieLens complete	0.6472	0.1104	82.9%
MovieLens (100 most popular tags)	0.925	0.223	75.9%
Facebook	0.4174	0.114	72.7%

TABLE 3.1: The densities of the projections and the backbones.

We extract the backbone of each of the above listed networks by determining the weight probability distribution for every edge between all possible pairs of nodes in each of the networks. An edge in the observed network is included in the backbone if its weight is significant, according to Definition 3.3. Determining edge significance by comparing each edge weight individually to its expected distribution ensures that edges with high weights are not chosen over edges with low weights, for inclusion in the backbone.

To illustrate that edge significance does not depend on edge weight, we have plotted their relationship in Figure 3.8 for the projection onto the senators and the projection onto the 100 most popular tags. The black horizontal line indicates the threshold of three standard deviations. Edge significance is calculated by subtracting the mean of the individual edge distribution from the observed weight and dividing by the distribution's standard deviation. This confirms that the weights of an edge do not determine its significance and hence a global threshold for edge removal would be inappropriate. Table 3.1 lists the densities of the projections and the densities of the backbones to show the reduction in size for each of the projections.

Extraction of the backbones of the different networks reveals the significant connections. As expected, these connections are found between similar nodes. In the projection of the U.S. Senate network onto the set of senators, the majority of edges in the backbone connect senators from the same party. Relatively few edges connect senators from different parties. Plots of the weight distributions of some of the most significant edges in the senator-senator projection as well as some of the edges that represent political antagonisms are depicted in Figures 3.9 and 3.10 respectively. Similar plots for the MovieLens network can be found in Appendix A (see Figures A.1 - A.2).

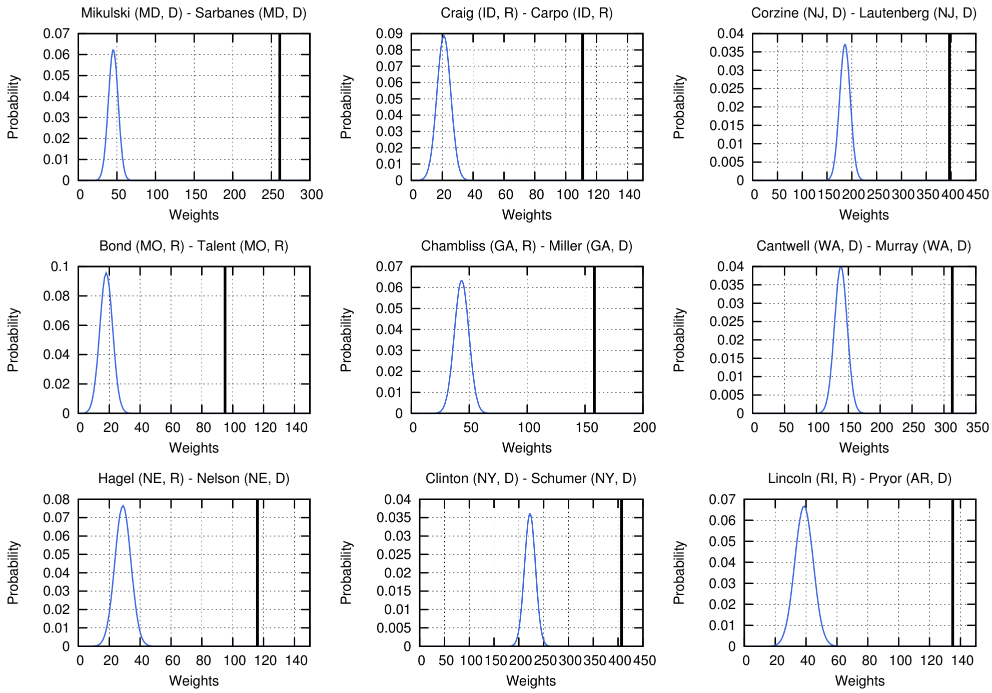


FIGURE 3.9: The weight probability distributions of the nine most significant edges in the senator-senator projection, where the observed weight is greater than expected. The blue curves show the approximated probability distributions, the black vertical bars mark the observed weight in the weighted one-mode projection of the 108$^{\mathrm{th}}$ U.S. Senate network.

Figure 3.11 shows the adjacency matrices of the binary projection, the weighted projection and the backbone of the different networks, with a black square indicating the presence of an edge. For the weighted projections, the different grey scales of the squares reflect the weight of the corresponding edges. All four backbones clearly show groups of highly connected nodes that are not visible in the binary and weighted projections. Networks that consist of groups of highly clustered nodes, with very few connections between the groups, are called networks with community structure (see Chapter 2, Subsection 2.2.4). Intuitively, if a network has community structure, this structure would be more pronounced in its backbone. This is especially the case for one-mode projections as the underlying communities may be hidden by noise. The following section demonstrates that backbone extraction aids in the detection of communities.

3.5 Detecting communities in one-mode projections

The previous section showed that the backbones of the examined real world networks consist of highly connected groups of nodes (see Figure 3.11). Thus, backbone extraction

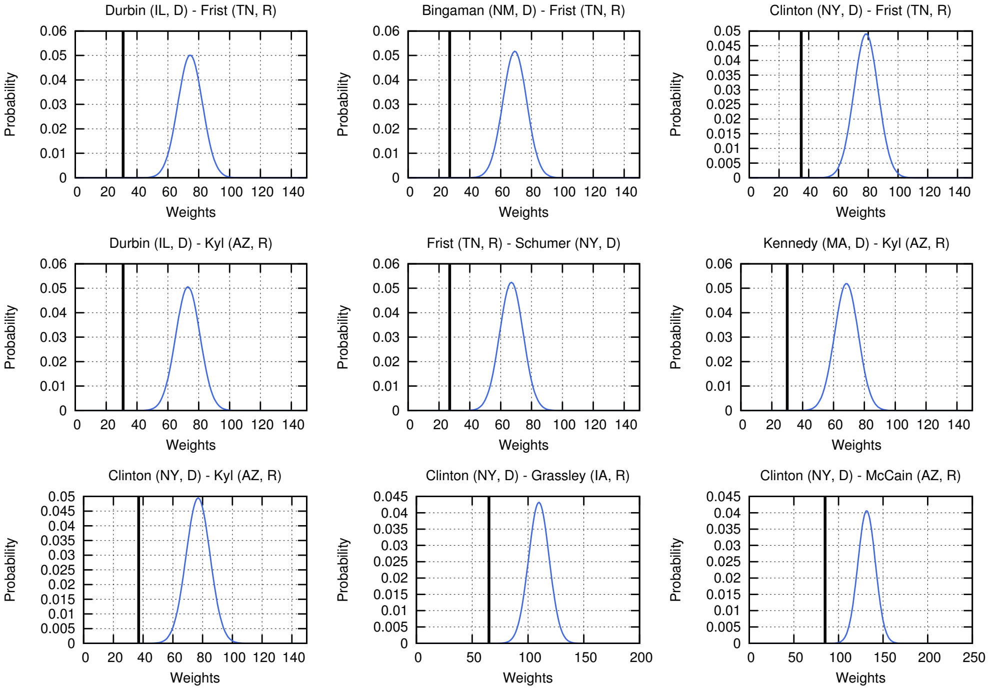


FIGURE 3.10: The weight probability distributions of nine edges in the senator-senator projection, where the observed weight is smaller than expected (not included in the backbone). These edges represent political antagonisms. The blue curves show the approximated probability distributions, the black vertical bars mark the observed weight in the weighted one-mode projection of the 108^{th} U.S. Senate network.

must remove inter community connections, while preserving links within communities. To show that our claim holds, we apply different community detection algorithms to the data sets examined in the previous section. A community is loosely defined as a sub-graph of a network with a relatively higher number of inner connections compared to the number of edges linking to nodes outside the sub-graph [12].

A lot of effort has been invested in the detection of network communities and a large body of literature exists on algorithms detecting them. The aim of running the different algorithms on real world data, is to show that in each case better results are achieved when inputting the backbone instead of the binary or weighted projection. We do not evaluate the individual performances of the community detection algorithms in this thesis.

There are many approaches to detecting communities, some of which are based on centrality measures, random walks, network flows or the spectrum of the network. We chose three popular algorithms (see Table 3.2) that have been implemented in the R programming language [103] by Csárdi and Nepusz [24] and used them to demonstrate that the communities of a network are easier to identify using its backbone.

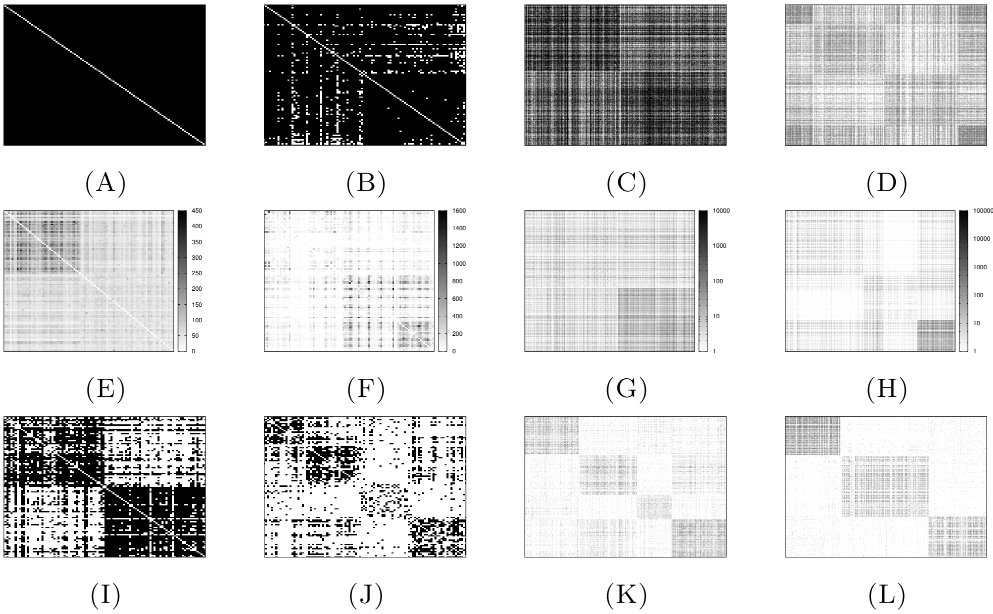


FIGURE 3.11: The adjacency matrices of the binary projections of the U.S. Senate network (A), the MovieLens network with the 100 most popular tags (B), the MovieLens network with all tags (C) and the Facebook network (D). The weighted projections of the U.S. Senate network (E), the MovieLens network with the 100 most popular tags (F), the MovieLens network with all tags (G) and the Facebook network (H). The backbones of the projections of the U.S. Senate network (I), the MovieLens network with the 100 most popular tags (J), the MovieLens network with all tags (K) and the Facebook network (L).

Algorithm	Approach	Reference
Louvain (see Subsection 2.2.4.2)	Modularity maximisation	[12]
Leading eigenvector (see Subsection 2.2.4.3)	Network spectrum	[85]
WalkTrap (see Subsection 2.2.4.4)	Random walks	[101]

TABLE 3.2: Three community detection algorithms that are based on different approaches.

To compare the performance of the algorithms with respect to the input networks (binary projection, weighted projection and backbone), we use the modularity function, given by Equation (2.5). The modularity function may be used to compare the partition of a network into communities achieved by algorithms that are not necessarily based on modularity maximisation (see Subsection 2.2.4). Note that the modularity function ranges between zero and one. The higher the modularity, the better the division of the network into groups of nodes.

3.5.1 108$^{\text{th}}$ U.S. Senate data

The 108$^{\text{th}}$ U.S. Senate network is known to contain two communities, democratic senate members and republican senate members [82]. Running the different community detection algorithms gives the results listed in Table 3.3. All three community detection algorithms achieved the highest modularity for the backbone. Since the binary projection is the complete graph $\mathcal{K}_{100}$ (see Figure 3.11A), all algorithms fail to detect the community structure. A list of the senators and their associated communities can be found in Appendix B (see Table B.1).

	Binary projection	Weighted projection	Backbone
Louvain [12]	0 (1)	0.0822 (2)	0.2367 (2)
Leading eigenvector [85]	0 (1)	0.0822 (2)	0.2367 (2)
WalkTrap algorithm [101]	0 (100)	0.0814 (2)	0.2239 (2)

TABLE 3.3: The modularities achieved by the different community detection algorithms. The value in the parenthesis indicates the number of detected communities. Note that if the WalkTrap algorithm fails to detect any communities it assigns each node to its own community, thus yielding 100 communities for the binary projection of the U.S. Senate network.

Newman's [85] and Blondel et al.'s [12] community detection algorithms achieved the best results. The two algorithms detected the exact same two communities (see Table B.1). 94% of the senators associated with the first community are democratic senators, whereas 96% of the senators associated with the second community are republican. The republican senators who are associated with the first community are Lincoln Chafee, Susan M. Collins and Olympia J. Snowe, whereas the democratic senators associated with the second community are Kent Conrad and Zell Miller.

Researching these senators revealed the following: Lincoln Chafee was a member of the Republican Party until 2007, when he became an independent, before joining the Democratic Party in 2013 [120]. Susan Collins, a known moderate member of the Republican Party, is considered bipartisan [116]. Like Collins, Olympia Snowe is also known to be strongly bipartisan [116]. Kent Conrad was found to be more conservative than most other democratic politicians [78], hence his association with the second community of mostly republican party members. Zell Miller was also found to be conservative [100]. In 2004 he backed President George W. Bush over the democratic nominee [8]. Figure 3.12 shows the backbone network with senator Miller and his neighbourhood highlighted.

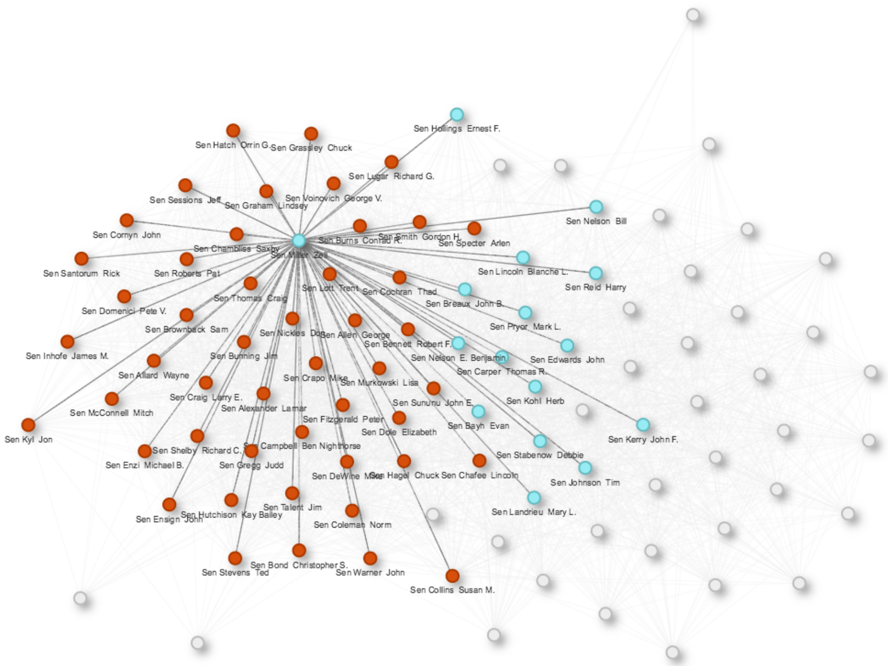


Figure 3.12: The backbone of the senator-senator projection with senator Zell Miller and his neighbourhood highlighted. Blue nodes represent democrats, red nodes represent republican members of the senate.

The neighbourhoods of the remaining four senators are depicted in Figures A.3 - A.6 in Appendix A.

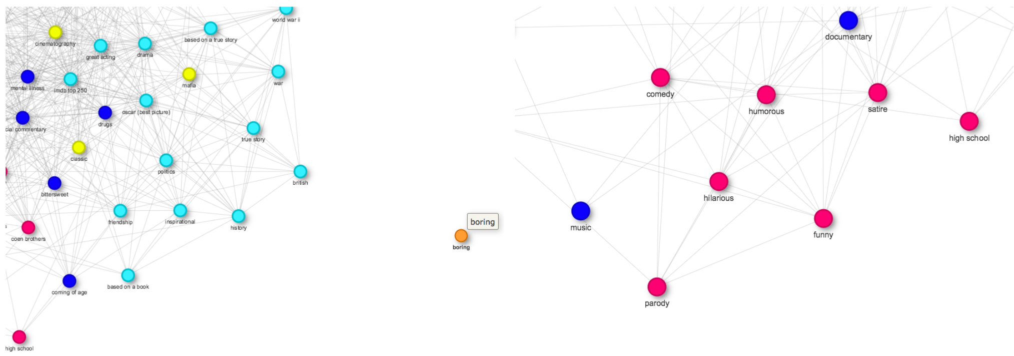
3.5.2 MovieLens Tags

Table 3.4 shows the results achieved by the different community detection algorithms for the binary tag-tag projection, the weighted projection and the backbone of the tag-tag projection.

	Binary projection top 100 tags/ all tags	Weighted projection top 100 tags/ all tags	Backbone top 100 tags/ all tags
Louvain [12]	0.0177 (2)/ 0.053 (6)	0.1321 (4)/ 0.13 (7)	0.2866 (6)/ 0.34 (10)
Leading eigenvector [85]	0.0178 (2)/ 0.049 (5)	0.1253 (3)/ 0.12 (6)	0.2608 (5)/ 0.32 (8)
WalkTrap algorithm [101]	0.0076 (2)/ 0 (1128)	0.1008 (2)/ 0.096 (5)	0.2306 (8)/ 0.3 (9)

Table 3.4: The modularities achieved by the different community detection algorithms. The value in the parenthesis indicates the number of detected communities.

Blondel et al.'s [12] community detection algorithms achieved the highest modularity and detected six communities in the backbone of the 100 most popular tags and ten communities in the backbone that includes all tags. A list of the 100 most popular tags and their community association found by the different algorithms in each of the three networks is given in Appendix B (see Table B.2).



(A) (B)

FIGURE 3.13: The backbone network of the tag-tag projection shows an isolated node that forms a community by itself. The different colours represent community membership (A). Looking closer at one of the communities in the backbone, we find that similar tags tend to form significant connections (B).

Interestingly, one of the the nodes in the backbone of the 100 most popular tags is isolated, forming a community by itself (see Figure 3.13A). This isolated tag, labelled *boring*, did not form any significant connections to other nodes in the network. The other six communities each contain tags that are very similar, for instance, the tags comedy, funny, humorous, satire and hilarious are members of the same community (see Figure 3.13B).

3.5.3 Facebook data

Table 3.5 shows the results achieved by the different community detection algorithms for the binary projection onto the set of candidates, the weighted projection and the backbone of the projection onto the set of candidates. Figure 3.14 displays the backbone of the projection onto the set of political candidates.

	Binary projection	Weighted projection	Backbone
Louvain [12]	0.11 (7)	0.31 (6)	0.6 (8)
Leading eigenvector [85]	0.086 (7)	0.29 (9)	0.56 (7)
WalkTrap algorithm [101]	0.066 (24)	0.26 (13)	0.6 (9)

TABLE 3.5: The modularities achieved by the different community detection algorithms. The value in the parenthesis indicates the number of detected communities.

The Louvain algorithm [12] identified eight communities in the backbone. Since four candidates are isolated, they each form a community by themselves. Community five consists mostly of members of the Australian Labour Party. Community six consists mostly of members of the Coalition (Country Liberal Party, Liberal National Party, Liberal Party of Australia and National Party of Australia). Community seven consists mostly of members of the Australian Greens. All remaining candidates are associated with the eighth community.

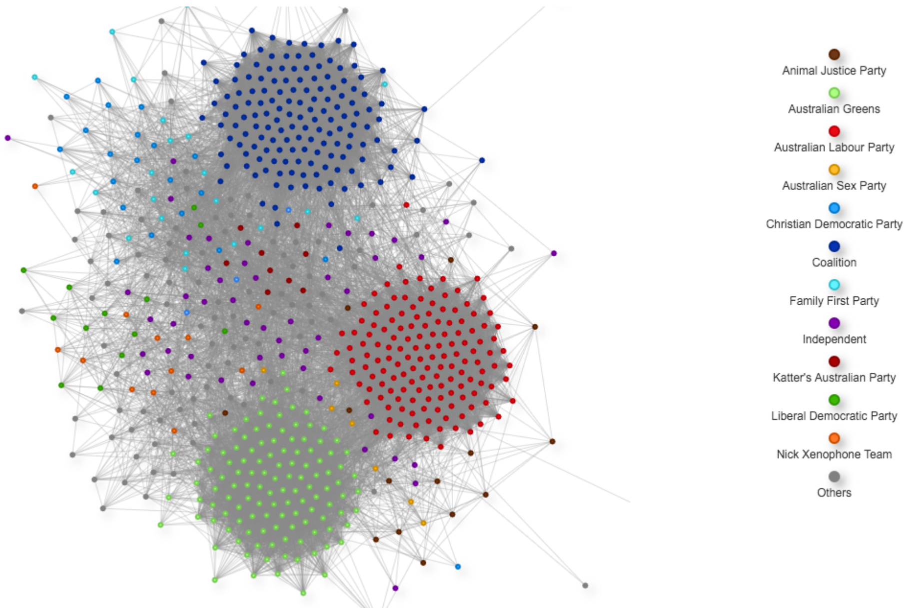


FIGURE 3.14: The backbone of the projection onto the set of political candidates. The nodes are coloured according to their party membership (see legend).

The leading eigenvector community detection algorithm [85] identified the same communities, with the difference that the candidates that Louvain associated with group eight are now each associated with one of the other seven communities.

The WalkTrap algorithm [101] identified nine communities in total. The communities were again the same as those identified by Louvain, with the difference that one candidate of one of the minor Australian parties now forms a community by herself.

Interestingly, for all three algorithms, two candidates that are members of one of the major Australian parties, were not associated with the community to which the other party members were assigned. David Atkins (Australian Labour Party) does not have any significant connections to other members of the Australian Labour Party. Mohit Kumar (Coalition) does not have many significant connections to other candidates in general. However, he has even less connections to members of the Coalition than to candidates of other parties.

3.5.4 Comparison to naive thresholds and one-mode methods

In this subsection we compare results obtained by our backbone extraction method to results obtained by using naive thresholds and one-mode backboning methods.

Figure 3.15 reports the results achieved by the different community detection algorithms for varying thresholds showing that removing edges with weights under a certain global threshold does not lead to an increase in modularity. Thus, naive edge removal leads to deletion of significant connections and therefore would not aid in the detection of communities. The figure clearly illustrates that trivial edge removal cannot achieve the same results as our backbone extraction method.

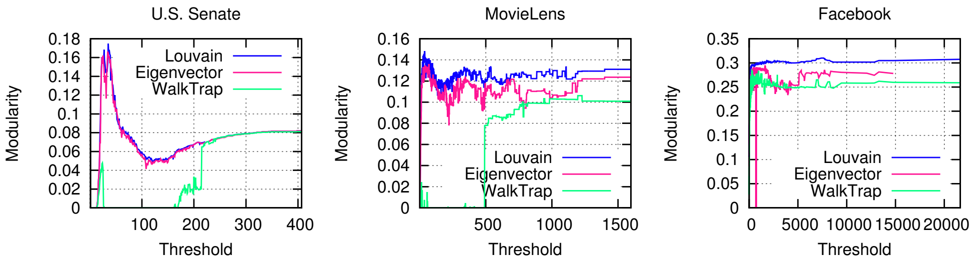


FIGURE 3.15: Removing edges with weights under a certain threshold demonstrates that an increase in modularity cannot be achieved by a trivial method. This figure presents a comparison of the modularities achieved by different community detection algorithms after removing edges by applying a global threshold for several real world datasets.

Similarly, when extracting the backbone using methods designed for weighted one-mode networks such as the one proposed in [39], significant connections tended to be deleted. We tested the extraction method of [39] on the projection of the U.S. Senate network and found that many edges between members of the same party were removed while many edges between members of different parties were retained. Running Newman's community detection algorithm [85] on the extracted backbone resulted in four communities all of them with a mixture of republicans and democrats.

The backbone extracted using our novel approach clearly leads to better detection results by the different community detection algorithms.

3.6 Summary

In this chapter we discussed several limitations of the one-mode projection, one of them being its high density. Projecting a bipartite network leads to an inflation of edges since connections in the projection are inferred indirectly. Hence not all connections in the projection are significant. This chapter dealt with the identification of significant connections in one-mode projections.

We introduced a novel technique for identifying the most significant connections in projected bipartite networks by showing that its edge weight distribution follows a Poisson binomial distribution. We tested the approximations to the weight distributions of several projections of random bipartite networks, leading to the conclusion that our approximation technique performs extremely well on sparse networks. The approximations are robust against variations of the network and degree distribution parameters.

Next, we extracted the backbones of several real world networks. The backbone extraction successfully revealed the underlying community structure of the projection that was previously not visible. Running different community detection algorithms on the binary projection, the weighted projection, and the backbone showed that the best results were achieved in the backbone network.

Chapter 4

The Clustering Coefficient

Parts of this chapter have been published in [66] and [67].

4.1 Introduction

4.1.1 Motivation

The clustering coefficient is a very important topological measure as it gives valuable insight into a network's structure by calculating the global concentration of triangles. Locally, it shows how well the neighbours of a particular node are connected to each other. The clustering coefficient is especially important in the analysis of social networks. In a friendship network, for example, the clustering coefficient measures the probability of two friends of a person being friends [86].

As a consequence of their particular structure, bipartite networks do not contain any cycles of odd length. Hence, the concentration of triangles and thus the clustering coefficient is zero in any network that has a bipartite structure. Consequently, the notion of clustering needs to be redefined to suit the analysis of bipartite networks.

4.1.2 Outline

Our contributions in this chapter are the following: We formally show that the global one-mode clustering coefficient of a projected bipartite network is generally higher than that of a similar random network. We define two different types of bipartite networks that differ in the way they develop over time. For each type of bipartite network we define bipartite clustering coefficients that are especially suited for their analysis.

The chapter is structured as follows: Section 4.2 provides the necessary background on the clustering coefficient of one-mode networks. Section 4.3 studies the clustering coefficient of one-mode projections and formally shows that the one-mode clustering coefficient of a projected bipartite network is generally higher than that of a similar random network. In Section 4.4 we critically examine previously proposed bipartite clustering coefficients and determine their limitations. We overcome these limitations by introducing novel bipartite clustering coefficients for different types of bipartite networks in Sections 4.5 and 4.6. We conclude this chapter with a summary in Section 4.7.

Applications of the different clustering coefficients are presented in Chapter 5.

4.2 The one-mode clustering coefficient

In a one-mode network, the clustering coefficient measures the concentration of triangles and hence, gives insight into the network's topology. For simple one-mode networks, i.e. undirected networks without self loops or multiple edges, the clustering coefficient cc is given by:

$$cc = \frac{3 \times \text{number of triangles}}{\text{total number of 2-paths}}. \tag{4.1}$$

In a simple one-mode network, a path of length two between two nodes u and u', that is, a path made up of two edges, contributes one to the total count of 2-paths. The factor of 3 in the numerator of Equation (4.1) accounts for the fact that every triangle contains exactly three paths of length two (see Figure 4.1).

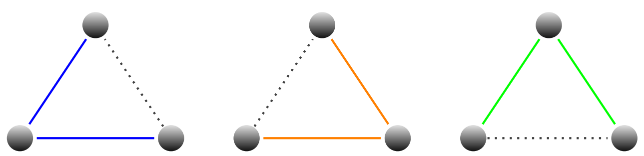


FIGURE 4.1: A triangle contains exactly three paths of length two.

Since Equation (4.1) gives the concentration of triangles in the whole network, it is referred to as the global clustering coefficient. The global clustering coefficient cc ranges from zero to one, where $cc = 1$ implies perfect transitivity. A clustering coefficient of zero implies that the network is triangle free [86].

Similar to calculating the global clustering coefficient of a network, one can measure the clustering coefficient of any vertex in a network. The clustering coefficient of a particular node u_i is called the local clustering coefficient of u_i. Instead of dividing the number of triangles by the total number of 2-paths, the number of triangles containing node u_i is divided by the total number of 2-paths centred at node u_i:

$$cc_i = \frac{\text{number of triangles containing node } u}{\text{number of 2-paths centred at node } u_i}. \tag{4.2}$$

Note that the global clustering coefficient cc is not the average of the local clustering coefficients cc_i.

Real world networks are often highly clustered compared to similar (with regards to order and size) random networks [83]. For example, in social networks the friends of a person have a high probability of being connected. Chen et al. [21] have shown that a node with a high local clustering coefficient is less effective in spreading a disease than a node with low local clustering coefficient. The clustering coefficients of projections were found to be especially high [94]. The following section investigates this.

4.3 Clustering in one-mode projections

Scientists often tend to project (see Definition 3.1) real world bipartite networks to calculate properties such as the clustering coefficient. When doing so, they usually find that the global clustering coefficient of the projection is much higher than that of a random one-mode network that has the same degree distribution or sequence as the projection [94]. This is due to the many cliques (a triangle is the smallest possible clique) that are created by projecting a bipartite network (see Subsection 3.2.1.2). This issue is well known amongst network scientists, however, it has not been formally shown that the clustering coefficient of a one-mode projection is generally higher than the clustering coefficient of a similar random network. To emphasise the necessity for a bipartite clustering coefficient, we formally confirm that the projection of a random bipartite network has a higher global clustering coefficient than a random one-mode network of the same size and order, and having the same degree distribution.

4.3.1 A general expression for the clustering coefficient of random one-mode networks

We start with an expression for the one-mode clustering coefficient of the configuration model (Subsection 2.2.5.1) in terms of the degree distribution of the network as presented in [86]. The configuration model is a random network model that fixes the order, size and degree sequence of the random network.

The one-mode clustering coefficient, as given by Equation (4.1), is equal to the average probability of two neighbours of a node being connected to each other. If u' and u'' are neighbours of u, the probability $\pi_{u'u''}$ of an edge linking nodes u' and u'' is given by:

$$\pi_{u'u''} = \frac{d_{u'} d_{u''}}{2m}, \tag{4.3}$$

where $d_{u'} = \deg(u') - 1$ and $d_{u''} = \deg(u'') - 1$ are the excess degrees of u' and u'' respectively [86]. The excess degrees, rather than simply the degrees, of the two nodes are multiplied, as both u' and u'' are connected to u. It is important to understand that $d_{u'}$ and $d_{u''}$ are not distributed according to the degree distribution of the network but follow the excess degree distribution. Newman [86] obtains the excess degree distribution in the configuration model by using generating functions. The necessary background on generating functions can be found in Chapter 2 (see Section 2.3).

The probability generating function of the degree distribution of a one-mode network is given by $h(x) = \sum_{d=0}^{\infty} r_d x^d$, where r_d is the fraction of nodes of degree d, and hence the number of nodes having degree d is equal to $|U| r_d$, where $|U|$ is the number of nodes in the network. Assuming that an edge is randomly chosen, the probability that it links to a node of degree d is $|U| d r_d / 2m = d r_d / \langle d \rangle$, where m is the number of edges in the network and $\langle d \rangle$ is the average node degree, i.e. the first moment of the degree distribution. Note that $d/2m$ is the probability of the edge ending at a node of degree d and $2m = |U| \langle d \rangle$ in a one-mode network.

The probability s_d of a randomly chosen edge attaching to a node that has excess degree d is then given by

$$s_d = \frac{(d+1)r_{d+1}}{\langle d \rangle}, \tag{4.4}$$

since the probability of a node having excess degree d is equal to the probability of that node having degree $d+1$.

With the above results, Newman [86] obtains a general expression for the clustering coefficient in the one-mode configuration model that depends only on the first and second moments of the degree distribution. The second moment of the degree distribution is denoted by $\langle d^2 \rangle$. Taking Equation (4.3) and summing $d_{u'}$ and $d_{u''}$ over the excess degree distribution s_d gives the clustering coefficient in one-mode networks:

$$\begin{aligned} cc &= \sum_{d_{u'},d_{u''}=0}^{\infty} \frac{d_{u'}d_{u''}}{2m} s_{d_{u'}} s_{d_{u''}} \\ &= \frac{1}{2m}\left[\sum_{d=0}^{\infty} d s_d\right]^2 \\ &= \frac{1}{2m}\left[\sum_{d=0}^{\infty} \frac{d(d+1)r_{d+1}}{\langle d \rangle}\right]^2 \\ &= \frac{1}{|U|\langle d \rangle^3}\left[\sum_{d=0}^{\infty} d(d-1)r_d\right]^2 \\ &= \frac{\left[\langle d^2 \rangle - \langle d \rangle\right]^2}{|U|\langle d \rangle^3}. \end{aligned} \tag{4.5}$$

This expression allows the calculation of the average clustering coefficient in the configuration model or a similar random network with a fixed degree distribution or sequence.

We generated several random one-mode networks, using the Curveball algorithm (see Section 2.2.5.2) to test how well Equation (4.5) estimates the clustering coefficient. We tested the following five degree distributions: The delta function, the uniform distribution, the normal distribution, the exponential distribution and the power law distribution. We varied the network and degree distribution parameters and generated 100 random networks for each variation. Figure 4.2 shows that Equation (4.5) estimates the clustering coefficient in random one-mode networks extremely well.

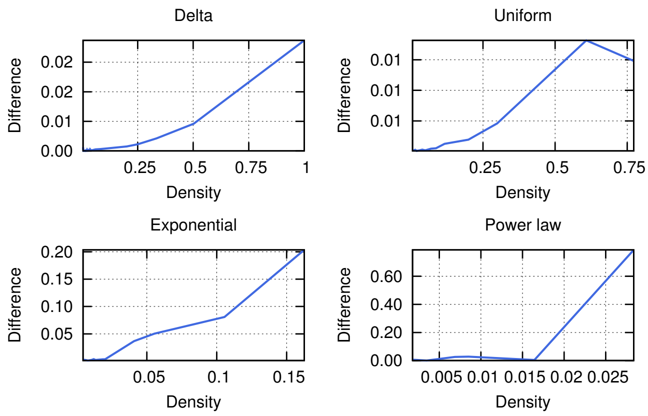


FIGURE 4.2: The expression given by Equation (4.5) approximates the clustering coefficient of the one-mode configuration model extremely well. The y-axes of the plots display the absolute difference between the observed clustering coefficient and the approximation.

4.3.2 Expressing the the clustering coefficient of projections in terms of moments

It is well known that the clustering coefficient of a projected bipartite network is generally higher than would be expected in a random one-mode network of the same order, size and with identical degree distribution. This indicates that Equation 4.5 turns out to be a poor estimator of the clustering coefficient in random bipartite networks. We confirmed this by generating several random bipartite networks, projecting them and calculating their global clustering coefficients (see Figure 4.3). Our results suggest that a random one-mode network is not suitable for comparison with a one-mode projection of a bipartite network and hence a better model is required. The random networks were generated as per Subsection 2.2.5.3, in Chapter 2.

The following subsection presents interesting background information on the relationship between the degree distributions of a bipartite network and the degree distribution of its weighted projection.

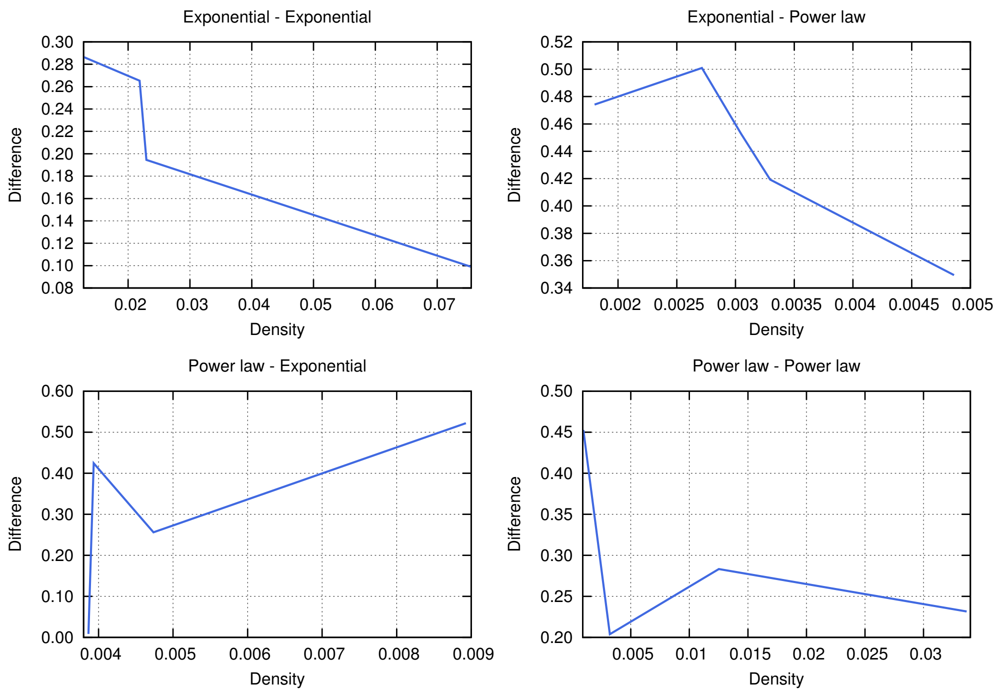


FIGURE 4.3: The expression given by Equation (4.5) poorly approximates the clustering coefficient of projections. The y-axes of the plots display the absolute difference between the observed clustering coefficient and the approximation. We considered the four most common combinations of degree distributions.

4.3.2.1 The generating function for the degree distribution of projections

We first give the expression for the generating function of the projection of a bipartite network as presented in [86]. Newman [86] has shown that the degree distribution of a weighted one-mode projection is determined by the degree distributions of the original bipartite network. The weighted one-mode projection of a bipartite network with biadjacency matrix B (see Definition 2.18) is given by BB^T, where B^T is the transpose of B. Calculating the probability of some node u of degree j in the bipartite network having degree d in the weighted one-mode projection is equivalent to calculating the probability that the excess degrees of its neighbours in the bipartite network add up to d. In what follows $f(x) = \sum_{j=0}^{\infty} p_j x^j$ is the probability generating function of the primary node set of the bipartite network, with p_j being the fraction of primary nodes of degree j. Similarly, the probability generating function of the secondary node set is given by $g(x) = \sum_{k=0}^{\infty} q_k x^k$.

Since the node u in the bipartite network has degree j, its j neighbours must together have d neighbours, not including u. Thus, the probability of j excess degrees taking a particular set of values $\{k_1, k_2, \ldots, k_j\}$ is $\prod_{v=1}^{j} s_{k_v}$. Summing over all sets $\{k_1, k_2, \ldots, k_j\}$, such that $k_1 + k_2 + \cdots + k_j = d$ results in the probability of the node u having degree d in the one-mode projection, given it has degree j in the bipartite network. Thus, if $h(x)$ denotes the generating function of the degree distribution of the one-mode projection, and r_d the fraction of nodes having degree d in the projection, then

$$\begin{aligned}
h(x) &= \sum_{d=0}^{\infty} r_d x^d \\
&= \sum_{d=0}^{\infty} x^d \sum_{j=0}^{\infty} p_j \sum_{k_1=0}^{\infty} \cdots \sum_{k_j=0}^{\infty} \delta\left(d, \sum_{v=1}^{j} k_v\right) \prod_{v=1}^{j} s_{k_v} \\
&= \sum_{j=0}^{\infty} p_j \sum_{k_1=0}^{\infty} \cdots \sum_{k_j=0}^{\infty} x^{\sum_{v=1}^{j} k_v} \prod_{v=1}^{j} s_{k_v} \\
&= \sum_{j=0}^{\infty} p_j \sum_{k_1=0}^{\infty} \cdots \sum_{k_j=0}^{\infty} \prod_{v=1}^{j} s_{k_v} x^{k_v} \\
&= \sum_{j=0}^{\infty} p_j \left[\sum_{k=0}^{\infty} s_k x^k\right]^j \\
&= \sum_{j=0}^{\infty} p_j \left[\sum_{k=0}^{\infty} \frac{(k+1)p_{k+1}}{\langle k \rangle} x^k\right]^j \\
&= \sum_{j=0}^{\infty} p_j \left[\frac{1}{\langle k \rangle} \sum_{k=1}^{\infty} k p_k x^{k-1}\right]^j \\
&= \sum_{j=0}^{\infty} p_j \left[\frac{1}{\langle k \rangle} g'(x)\right]^j \\
&= f\left(\frac{1}{\langle k \rangle} g'(x)\right), \qquad (4.6)
\end{aligned}$$

where $\delta(a, b)$ is the Kronecker delta, with $\delta(a, b) = 1$ if $a = b$ and 0 otherwise.

We next calculate the clustering coefficient of a random one-mode network with the same degree distribution as the projection of a given bipartite network. Using the above result, we give an expression in terms of the first and second moments of the degree distributions of the bipartite network that allows us to measure the clustering coefficient of a random one-mode network with the same degree distribution as the projection of a bipartite network. This expression makes the process of projecting the bipartite network and the subsequent randomisation of the projection unnecessary. We note here that we have been unable to find this expression in the literature.

4.3.2.2 An expression for the clustering coefficient of one-mode random networks with the same degree distribution as a projection

The first moment of the degree distribution, i.e. the average node degree, of the projection of a bipartite network is given by

$$\begin{aligned}
\langle d \rangle &= h'(1) \\
&= \left[\frac{d}{dx} \left[f \left(\frac{1}{\langle k \rangle} g'(x) \right) \right] \right]_{x=1} \\
&= \left[\frac{1}{\langle k \rangle} f' \left(\frac{1}{\langle k \rangle} g'(x) \right) g''(x) \right]_{x=1} \\
&= \frac{1}{\langle k \rangle} f' \left(\frac{1}{\langle k \rangle} g'(1) \right) g''(1) \\
&= \frac{1}{\langle k \rangle} f'(1) g''(1) \\
&= \frac{\langle j \rangle}{\langle k \rangle} \sum_{k=0}^{\infty} k(k-1) q_k \\
&= \frac{\langle j \rangle \left[\langle k^2 \rangle - \langle k \rangle \right]}{\langle k \rangle}. \qquad (4.7)
\end{aligned}$$

The second moment of the degree distribution of the projection of a bipartite network is given by

$$\begin{aligned}
\langle d^2 \rangle &= \left[\left(x \frac{d}{dx} \right)^2 h(x) \right]_{x=1} \\
&= \left[x \frac{d}{dx} \left(x \frac{dh}{dx} \right) \right]_{x=1} \\
&= \left[x \frac{d}{dx} \left(x \frac{d}{dx} \left(f \left(\frac{1}{\langle k \rangle} g'(x) \right) \right) \right) \right]_{x=1} \\
&= \left[x \frac{d}{dx} \left(\frac{x}{\langle k \rangle} f' \left(\frac{1}{\langle k \rangle} g'(x) \right) g''(x) \right) \right]_{x=1}
\end{aligned}$$

$$
\begin{aligned}
&= \left[x\left(\frac{1}{\langle k\rangle}f'\left(\frac{1}{\langle k\rangle}g'(x)\right)g''(x)+\frac{x}{\langle k\rangle^2}f''\left(\frac{1}{\langle k\rangle}g'(x)\right)g''(x)g''(x)\right.\right.\\
&\quad \left.\left.+\frac{x}{\langle k\rangle}f'\left(\frac{1}{\langle k\rangle}g'(x)\right)g'''(x)\right)\right]_{x=1}\\
&= \frac{1}{\langle k\rangle}f'(1)g''(1)+\frac{1}{\langle k\rangle^2}f''(1)g''(1)g''(1)+\frac{1}{\langle k\rangle}f'(1)g'''(1)\\
&= \frac{\langle j\rangle}{\langle k\rangle}\sum_{k=0}^{\infty}k(k-1)q_k+\frac{1}{\langle k\rangle^2}\sum_{j=0}^{\infty}j(j-1)p_j\left[\sum_{k=0}^{\infty}k(k-1)q_k\right]^2\\
&\quad +\frac{\langle j\rangle}{\langle k\rangle}\sum_{k=0}^{\infty}k(k-1)(k-2)q_k\\
&= \frac{\langle j\rangle\left[\langle k^2\rangle-\langle k\rangle\right]}{\langle k\rangle}+\frac{\left[\langle j^2\rangle-\langle j\rangle\right]\left[\langle k^2\rangle-\langle k\rangle\right]^2}{\langle k\rangle^2}\\
&\quad +\frac{\langle j\rangle\left[\langle k^3\rangle-3\langle k^2\rangle+2\langle k\rangle\right]}{\langle k\rangle}
\end{aligned}
\tag{4.8}
$$

Equation (4.5) can now be written as

$$
\begin{aligned}
cc &= \frac{\left[\langle d^2\rangle-\langle d\rangle\right]^2}{|U|\langle d\rangle^3}\\
&= \left[\frac{\left[\langle j^2\rangle-\langle j\rangle\right]\left[\langle k^2\rangle-\langle k\rangle\right]^2}{\langle k\rangle^2}+\frac{\langle j\rangle\left[\langle k^3\rangle-3\langle k^2\rangle+2\langle k\rangle\right]}{\langle k\rangle}\right]^2 \Big/ |U|\frac{\langle j\rangle^3\left[\langle k^2\rangle-\langle k\rangle\right]^3}{\langle k\rangle^3}\\
&= \frac{\left[\langle j^2\rangle-\langle j\rangle\right]^2\left[\langle k^2\rangle-\langle k\rangle\right]}{|U|\langle j\rangle^3\langle k\rangle}+\frac{\langle k\rangle\left[\langle k^3\rangle-3\langle k^2\rangle+2\langle k\rangle\right]^2}{|U|\langle j\rangle\left[\langle k^2\rangle-\langle k\rangle\right]^3}\\
&\quad +\frac{2\left[\langle j^2\rangle-\langle j\rangle\right]\left[\langle k^3\rangle-3\langle k^2\rangle+2\langle k\rangle\right]}{|U|\langle j\rangle^2\left[\langle k^2\rangle-\langle k\rangle\right]}.
\end{aligned}
\tag{4.9}
$$

Equation (4.9) is an expression for the clustering coefficient in a random one-mode network that has the same degree distribution as the projection of a bipartite network with primary degree distribution $f(x) = \sum_{j=0}^{\infty} p_j x^j$ and secondary degree distribution $g(x) = \sum_{k=0}^{\infty} p_k x^k$.

Since Equation (4.5) is an approximation of the unweighted global clustering coefficient in one-mode networks, the expression given by Equation (4.9) gives good approximations only for very sparse bipartite networks. The reason is that the original expression given by Equation (4.5) is for simple graphs, graphs without loops and multiple edges. Hence

Equation (4.9) should only be used for binary projections. Denser bipartite networks would result in projections with multiple edges. Note that for sparse bipartite networks, the probability of an edge in the projection having a weight greater than one is very low. Hence in the case of sparse bipartite networks the weighted projection is highly likely to be binary.

The next subsection formally shows that the clustering coefficient of a one-mode projection is higher than the clustering coefficient of a random one-mode network with the same degree distribution.

4.3.2.3 The clustering coefficient of a projected bipartite network is higher than expected

By using Equation (4.5) to calculate the clustering coefficient in a random one-mode network, it is assumed that the two nodes u' and u'' are connected to node u (see Figure 4.4A). If the one-mode network is a projection, this implies that the structure depicted in Figure 4.4B exists in the bipartite network. However, it may be the case that nodes v and v' are the same node (see Figure 4.4C). This possibility is ignored by Equation (4.5).

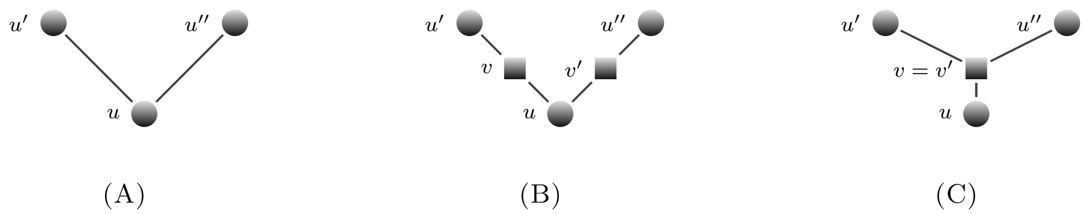


FIGURE 4.4: The induced sub-graph of a one-mode projection depicted in (A) can arise from only one bipartite sub-graph, that depicted in (B). If the two secondary nodes v and v' are the same node (C), the probability of u' and u'' being connected in one.

We can now formally show that the clustering coefficient of the projection of a random bipartite network is higher than the clustering coefficient of a random one-mode network of the same order, size and with the same degree distribution as the projection, by finding the probability of two second neighbours of u, say u' and u'', already being connected via a first neighbour of u, that is the case where $v = v'$. Note that the probability that u' and u'' are connected, given that $v = v'$, is equal to one.

To calculate the probability that any two second neighbours of node u are already connected via a first neighbour of u, one needs to find the average number of second neighbours per first neighbours of u. This number is equal to the average excess degree of a first neighbour of u, and hence equal to

$$\begin{aligned}\sum_{k=0}^{\infty} \frac{k(k+1)q_{k+1}}{\langle k \rangle} &= \frac{1}{\langle k \rangle} \sum_{k=0}^{\infty} k(k-1)q_k \\ &= \frac{\langle k^2 \rangle - \langle k \rangle}{\langle k \rangle}. \end{aligned} \tag{4.10}$$

It follows that the total number of second neighbours is given by $\langle j \rangle \left[\langle k^2 \rangle - \langle k \rangle\right] / \langle k \rangle$.

The number of pairs of second neighbours that are already connected is then given by $\langle j \rangle \binom{\frac{\langle k^2 \rangle - \langle k \rangle}{\langle k \rangle}}{2}$ and the total number of possible pairs amongst the second neighbours is equal to $\binom{\frac{\langle j \rangle \left[\langle k^2 \rangle - \langle k \rangle\right]}{\langle k \rangle}}{2}$.

Hence, the probability of any two second neighbours of node u already being connected through one of u's first neighbours is equal to

$$\begin{aligned} P(v = v') &= \langle j \rangle \binom{\frac{\langle k^2 \rangle - \langle k \rangle}{\langle k \rangle}}{2} \Big/ \binom{\frac{\langle j \rangle \left[\langle k^2 \rangle - \langle k \rangle\right]}{\langle k \rangle}}{2} \\ &= \frac{\langle j \rangle \left[\frac{\langle k^2 \rangle - \langle k \rangle}{\langle k \rangle}\right]!}{2! \left[\frac{\langle k^2 \rangle - \langle k \rangle}{\langle k \rangle} - 2\right]!} \frac{2! \left[\langle j \rangle \frac{\langle k^2 \rangle - \langle k \rangle}{\langle k \rangle} - 2\right]!}{\left[\langle j \rangle \frac{\langle k^2 \rangle - \langle k \rangle}{\langle k \rangle}\right]!} \\ &= \frac{\langle j \rangle \left[\frac{\langle k^2 \rangle - \langle k \rangle}{\langle k \rangle}\right] \left[\frac{\langle k^2 \rangle - \langle k \rangle}{\langle k \rangle} - 1\right]}{\left[\langle j \rangle \frac{\langle k^2 \rangle - \langle k \rangle}{\langle k \rangle}\right] \left[\langle j \rangle \frac{\langle k^2 \rangle - \langle k \rangle}{\langle k \rangle} - 1\right]} \\ &= \frac{\left[\frac{\langle k^2 \rangle - \langle k \rangle}{\langle k \rangle} - 1\right]}{\left[\langle j \rangle \frac{\langle k^2 \rangle - \langle k \rangle}{\langle k \rangle} - 1\right]} \\ &= \frac{\langle k^2 \rangle - 2\langle k \rangle}{\langle j \rangle \left[\langle k^2 \rangle - \langle k \rangle\right] - \langle k \rangle}. \end{aligned} \tag{4.11}$$

Since $\langle k^2 \rangle > \langle k \rangle$ and given that we have randomly chosen two second neighbours of node u, the probability that these two neighbours are already connected via a first neighbour of u is clearly a positive quantity (see Equation (4.11)) that is not taken into consideration when calculating the clustering coefficient of a random one-mode network that has the same degree distribution as the projection of a given bipartite network.

Thus we have formally calculated the probability of two of u's second neighbours already being connected via a first neighbour of u, demonstrating that the clustering coefficient in projections is generally higher than in similar random one-mode networks.

4.4 The bipartite clustering coefficient

The previous section clearly showed that the clustering coefficient of the projection of a random bipartite network is higher than the clustering coefficient of a random one-mode network that follows the same degree distribution as the projection. When calculating the clustering coefficient of the one-mode projection of a particular bipartite network, it is therefore advisable to randomise the bipartite network, then project and calculate its clustering coefficient for comparison. Since one-mode projections are computationally expensive, this process becomes infeasible very quickly as the bipartite network grows. One-mode projections are matrix multiplications, running in $\mathcal{O}(|U|^2|V|)$ time. Hence, a definition of the clustering coefficient applicable specifically to bipartite networks would be highly beneficial for their analysis, as it would reveal much about the topology of bipartite networks.

Several definitions of the bipartite clustering coefficient have been proposed [70, 94, 109, 134]. These definitions are, however, inconsistent and hence require further investigation. Whereas most authors measure the concentration of 4-cycles, Opsahl [94] was the first to consider cycles of length six. This section reviews some of the existing bipartite clustering coefficients and points out their limitations.

4.4.1 Concentration of 4-cycles

Most of the proposed bipartite clustering coefficients measure the concentration of squares, that is cycles of length four, in the network of interest. The authors usually argue that a triangle is the smallest possible cycle in a one-mode network and since a square is the smallest possible cycle in a bipartite network, their concentration should be measured instead. However, even here, the way in which the concentration is measured often differs.

4.4.1.1 Robins et al.'s clustering coefficient

Robins and Alexander [109] calculate the concentration of 4-cycles by dividing four times their number by the number of paths of length three and hence

$$cc_{\text{bip}} = \frac{4 \times \text{number of 4-cycles}}{\text{number of 3-paths}}. \tag{4.12}$$

The bipartite clustering coefficient given by Equation (4.12) shows how likely two, say primary, nodes are connected multiple times via different secondary nodes. Hence, a high clustering coefficient implies high connectivity between pairs of nodes of the same type.

Robins and Alexander [109] investigate networks of corporate boards and directors in the United States and Australia and are interested in the level of connectivity between any two directors, making their clustering coefficient a useful tool. On the other hand, their clustering coefficient fails to determine how well any three nodes of the same type are connected to each other, as intended by the one-mode clustering coefficient.

4.4.1.2 Lind et al.'s clustering coefficient

Lind et al. [70] propose a local bipartite clustering coefficient that, similar to Equation (4.12), measures the concentration of 4-cycles, with the concentration calculated differently. Consider the network depicted in Figure 4.5. The local clustering coefficient of node u is obtained by dividing the number of 4-cycles containing node u by the number of all possible 4-cycles plus the 4-cycles containing u. Lind et al. [70] define a possible 4-cycle as the possible overlap of second neighbours of u. For instance, the path of length four between nodes u' and u'' via node u is considered a possible 4-cycle, as an overlap of u' and u'' would form a cycle of length four. Formally,

$$cc_{\text{bip}}(u) = \sum_{v<v'} \frac{q_{uvv'}}{(k_v - 1 - q_{uvv'})(k_{v'} - 1 - q_{uvv'}) + q_{uvv'}}, \tag{4.13}$$

where $q_{uvv'}$ is the number of 4-cycles containing nodes u, v and v' and $\deg(v) = k_v$.

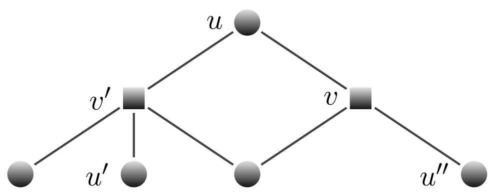


FIGURE 4.5: A small bipartite network. Lind et al. [70] define a possible 4-cycle as the possible overlap of second neighbours of u. Zhang et al. [134] on the other hand, define a possible 4-cycle as a path of length three that can be closed to form a 4-cycle by connecting a secondary node to the two primary end nodes of a 3-path.

The definition of possible 4-cycles, given in [70] is questionable. According to Lind et al.'s [70] definition, a possible 4-cycle is equivalent to a path of length four. It is worth noting that a path of length four cannot form a cycle of length four unless the nodes at the end of the path are allowed to merge.

4.4.1.3 Zhang et al.'s clustering coefficient

Zhang et al. [134] propose a clustering coefficient that is very similar to the one suggested in [70], with the two coefficients differing in the definition of a possible 4-cycle. Whereas Lind et al. [70] consider the possible overlap of nodes, Zhang et al. [134] consider possible edges that may form to create a 4-cycle, in other words, a possible 4-cycle is equivalent to a path of length three. Formally,

$$cc_{\text{bip}}(u) = \sum_{v<v'} \frac{q_{uvv'}}{(k_v - 1 - q_{uvv'}) + (k_{v'} - 1 - q_{uvv'}) + q_{uvv'}}, \tag{4.14}$$

where $q_{uvv'}$ is the number of 4-cycles containing nodes u, v and v' and $k_v = \deg(v)$.

Like the clustering coefficient that is given by Equation (4.12), the clustering coefficients given in [70] and in [134] measure the level of connectivity between any two nodes of the same type. Although all three measures give insights into a network's structure, calling these measures clustering coefficients is misleading, as the original clustering coefficient measures transitivity and therefore closure between three nodes. The first person to measure closure between three nodes of the same type in bipartite networks is Opsahl [94]. His way of measuring the bipartite clustering coefficient is discussed in the next subsection.

4.4.2 Concentration of 6-cycles

The bipartite clustering coefficient introduced by Opsahl [94] is very different to those discussed in Subsection 4.4.1, as it measures the concentration of 6-cycles in bipartite networks. Opsahl [94] argues that the bipartite clustering coefficient should do the same as the one-mode clustering coefficient, that is, measure triadic closure.

There are two candidate sub-graphs that could be considered a closed connection between three nodes in a bipartite network, a cycle of length six and a 3-star (see Figure 4.6). In both sub-graphs all three primary nodes are indirectly connected to each other and hence result in triangles when projected onto a one-mode network (see Chapter 3, Figure 3.2).

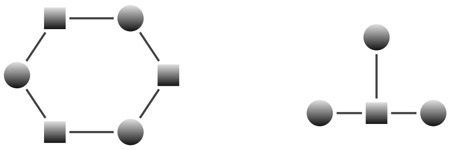


FIGURE 4.6: Both sub-graphs may be considered a closed connection between the three primary nodes.

Connecting a node to the two nodes at the ends of a 4-path, as depicted in Figure 4.7, forms a closed connection between three nodes, namely a 6-cycle [94]. As a similar kind of formation is not possible for a star sub-graph and since star sub-graphs are the reason for a higher than expected clustering coefficient in one-mode projections (see Subsection 4.3.2), Opsahl [94] chooses not to consider these sub-graphs as closed. He gives the following equation to calculate clustering in bipartite networks:

$$cc_{\text{bip}} = \frac{\tau_{\triangle}}{\tau}, \tag{4.15}$$

where τ is the number of 4-paths and $\tau_{\triangle}$ is the number of these that are closed and form a 6-cycle.

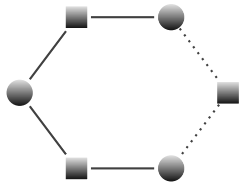


FIGURE 4.7: Connecting a secondary node to the two primary nodes at the end of a 4-path forms a cycle of length six.

The clustering coefficient given in [94] is the first step towards analysing triadic closure in bipartite networks. However, there are two major limitations. Firstly, the clustering coefficient given by Equation (4.15) does not take different types of 6-cycles into consideration. Secondly, Equation (4.15) assumes a certain manner in which the bipartite network develops over time. We address both issues in the following two subsections respectively.

4.4.3 Structures of bipartite clusters

This section examines the different structures that a bipartite 6-cycle may have. We give a mathematical proof, showing that failing to distinguish between these structures leads to an over count of 6-cycles.

In a bipartite network, the primary and secondary nodes of a 6-cycle may be connected to each other by additional edges (see Figure 4.8). These additional edges are called chords (see Definition 2.15), giving rise to differently structured 6-cycles with distinct meanings that depend on the network in question. In Chapter 5 we consider the different 6-cycles in the concrete context of real world networks and the reasoning for these differently structured 6-cycles will become more obvious.

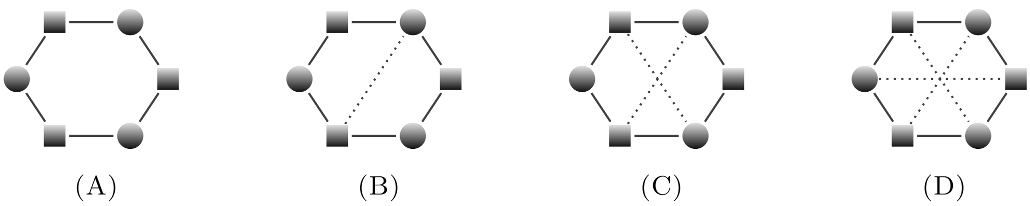


FIGURE 4.8: A bipartite 6-cycle may have a maximum of three chords, resulting in four differently structured clusters: (A) an induced 6-cycle, (B) a 6-cycle with one chord, (C) a 6-cycle with two chords and (D) a 6-cycle with three chords. Chords are represented by dashed lines.

By traversing along different edges of the four structures shown in Figure 4.8, one can confirm that an induced 6-cycle contributes one to the overall count of 6-cycles. A 6-cycle with one chord contains a single 6-cycle and hence also contributes one to the overall count of 6-cycles. A 6-cycle with two chords contributes two to the overall count of 6-cycles and finally, a 6-cycle with three chords contributes six to the overall count of 6-cycles. This shows the importance of distinguishing between the different structures, otherwise 6-cycles are over-counted. Following, we give a formal proof:

Theorem 4.1. *An induced 6-cycle contributes one to the overall count of 6-cycles. A 6-cycle with one chord contributes one to the overall count of 6-cycles. A 6-cycle with two chords contributes two to the overall count of 6-cycles, and a 6-cycle with three chords contributes six to the overall count of 6-cycles.*

Proof. Note that the direction of the 6-cycle does not matter. In addition, the cycle can start at any node.

Let $\mathcal{B}$ be a bipartite network and B its biadjacency matrix. Let $C_6 = \{u_i, v_r, u_j, v_s, u_k, v_t, u_i\}$ be a 6-cycle in $\mathcal{B}$ that may or may not have any chords. A bipartite 6-cycle has three distinct primary and three distinct secondary nodes.

Every node that is part of a 6-cycle must have degree at least two and exactly two of the incident edges must be part of the cycle. Any 6-cycle in $\mathcal{B}$ corresponds to a 3×3 submatrix of B that contains at least two ones in every row and every column.

The cycle C_6 may thus be represented by $S = \begin{bmatrix} 1 & b_{is} & 1 \\ 1 & 1 & b_{jt} \\ b_{kr} & 1 & 1 \end{bmatrix}$. The entries b_{is}, b_{jt} and b_{kr} are not part of C_6 and may equal zero or one. Note that since it does not matter which row represents which primary node and which column represents which secondary node, the rows and columns of S can be rearranged to give $S = \begin{bmatrix} b_{is} & 1 & 1 \\ 1 & b_{jt} & 1 \\ 1 & 1 & b_{kr} \end{bmatrix}$, placing the entries that are not part of C_6 in the diagonal of the matrix.

There are then four possibilities: i) S contains exactly three zero entries, ii) S contains exactly two zero entries, iii) S contains exactly one zero entry or iv) S does not contain any zero entries. In order to find the number of 6-cycles that S represents, all entries that are equal to zero are dropped from consideration as they represent non-existing edges. Every node that is part of a 6-cycle has two incident edges that are also part of the cycle. Thus, if an entry of S is dropped from consideration, all other entries in the corresponding row and column must be part of the 6-cycle. From the four possibilities above, we get:

i) All entries that are equal to zero are dropped. The entries in the corresponding rows and columns must be part of the 6-cycle and hence S represents exactly one 6-cycle.

ii) Without loss of generality, assume that $b_{is} = b_{jt} = 0$ and $b_{kr} = 1$. The two entries that are equal to zero are dropped. The entries in the corresponding rows and columns must be part of the 6-cycle and hence S represents exactly one 6-cycle.

iii) Without loss of generality, assume that $b_{is} = 0$ and $b_{jt} = b_{kr} = 1$. The entry that is equal to zero is dropped, and the entries in the corresponding row and column must be part of the 6-cycle. We can then drop any one of the two remaining entries in the second column, thus giving $\begin{pmatrix} 2 \\ 1 \end{pmatrix} \begin{pmatrix} 1 \\ 1 \end{pmatrix} = 2$ cycles of length six.

iv) Any one of the three entries in the first column is dropped from consideration, leading to the entries in the corresponding row and column being part of the 6-cycle. Next, any one of the two remaining entries in the second column is dropped, thus giving $\begin{pmatrix} 3 \\ 1 \end{pmatrix} \begin{pmatrix} 2 \\ 1 \end{pmatrix} = 6$ cycles of length six.

$\square$

Theorem 4.1 underlines the importance of distinguishing between the four cycles shown in Figure 4.8 in order to avoid an over-count that would lead to false results. For instance, a 6-cycle with two chords contributes two to the overall count of 6-cycles. However, there can only be one closed connection between three nodes of the same type. Distinguishing between the different structures also gives insight into the interconnectedness of any set of three nodes of the same type.

4.4.4 Formation of clusters

The way in which the different clusters, depicted in Figure 4.8, are formed, depends very much on the nature of the network. Consider a network that models the attendance of people at events, where a person can attend an event only at the time it takes place. At each time step an event is added to the network. At the same time people form links to this event by attending it. According to this example, only a path of length four can form a cycle of length six in the following time step. Equation (4.15) indirectly assumes exactly this.

Rating networks, on the other hand develop very differently, violating the assumption made by Equation (4.15). In a rating network users form edges to items by rating them. Since an item may be rated at any time, a cycle of length six could be formed by adding an edge between the nodes at the end of a path of length five.

The difference in formation of the two examples above requires different clustering coefficients, accounting for their development over time. In the following two sections we introduce different clustering coefficients for each of the two types of bipartite networks. Henceforth, we call the first type a time dependent bipartite network as links to a particular node, say an event, can only be formed at a particular point in time. The second type is called a time independent bipartite network, as edges may be formed at any point in time.

4.5 A clustering coefficient for time dependent networks

In Subsection 4.4.3 we pointed out the importance of separating the differently structured 6-cycles. Hence, we do not introduce a single, but four distinct clustering coefficients that correspond to the different 6-cycles depicted in Figure 4.8. Before defining the four clustering coefficients, a clear understanding of their formation is needed.

In a time dependent network, a secondary node is added at each time step, with primary nodes forming edges to the added node at that particular time. Figure 4.9 shows the different possibilities by which the different 6-cycles may be formed in time dependent networks.

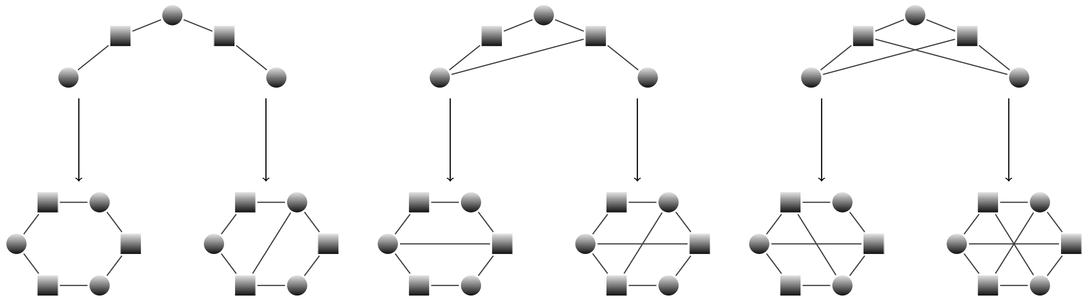


FIGURE 4.9: All possibilities by which the differently structured 6-cycles can be formed in a time dependent network. An induced 6-cycle and a 6-cycle with three chords can each only originate from one, distinct 4-path. 6-cycles with one or two chords each have two origins. (A) A 4-path without any additional edges is called an induced 4-path. (B) A 4-path with one additional edge is called a 4-path with one chord. (C) A 4-path with two additional edges is called a 4-path with two chords.

It is important to notice that once a 6-cycle is formed in a time dependent bipartite network, it cannot change its structure. This is however possible in time independent networks.

Using the origins of 6-cycles that are depicted in Figure 4.9, we define Equations (4.16) - (4.19) to enable the measurement of four different bipartite clustering coefficients dcc_x.

The induced clustering coefficient:

$$dcc_0 = \frac{\lambda_0^*}{\lambda_0}, \tag{4.16}$$

where λ_0^* is the number of closed 4-paths that form an induced 6-cycle and λ_0 the total number of induced 4-paths. The induced clustering coefficient dcc_0 measures the proportion of induced 4-paths that are closed, thus forming an induced 6-cycle.

The one chord clustering coefficient:

$$dcc_1 = \frac{\lambda_1^*}{\lambda_0 + \lambda_1}, \tag{4.17}$$

where λ_1^* is the number of closed 4-paths that form a 6-cycle with one chord and λ_1 the total number of 4-paths with one chord. The one chord clustering coefficient dcc_1 measures the proportion of 4-paths that are closed, thus forming a 6-cycle with one chord.

The two chord clustering coefficient:

$$dcc_2 = \frac{\lambda_2^*}{\lambda_1 + \lambda_2}, \tag{4.18}$$

where λ_2^* is the number of closed 4-paths that form a 6-cycle with two chords and λ_2 the total number of 4-paths with two chords. The two chord clustering coefficient dcc_2 measures the proportion of 4-paths that are closed, thus forming a 6-cycle with two chords.

The three chord clustering coefficient:

$$dcc_3 = \frac{\lambda_3^*}{\lambda_2}, \tag{4.19}$$

where λ_3^* is the number of closed 4-paths that form a 6-cycle with three chords. The clustering coefficient dcc_3 measures the proportion of 4-paths that are closed and form a 6-cycle with three chords.

The local clustering coefficients $dcc_{i,x}$ of a node u_i can be measured in a similar manner. For example, the local clustering coefficient $dcc_{i,0}$ of the node u_i is measured by dividing the number of closed 4-paths that are centred at u_i and form an induced 6-cycle by the number of all induced 4-paths that are centred at u_i. The local clustering coefficients are given by the following four equations:

The induced local clustering coefficient:

$$dcc_{i,0} = \frac{\lambda_{i,0}^*}{\lambda_{i,0}}, \tag{4.20}$$

where $\lambda_{i,0}^*$ is the number of closed 4-paths, centred at node u_i, that form an induced 6-cycle and $\lambda_{i,0}$ the total number of induced 4-paths that include node u_i.

The one chord local clustering coefficient:

$$dcc_{i,1} = \frac{\lambda_{i,1}^*}{\lambda_{i,0} + \lambda_{i,1}}, \tag{4.21}$$

where $\lambda^*_{i,1}$ is the number of closed 4-paths, centred at node u_i, that form a 6-cycle with one chord and $\lambda_{i,1}$ the total number of 4-paths with one chord that include node u_i.

The two chord local clustering coefficient:

$$dcc_{i,2} = \frac{\lambda^*_{i,2}}{\lambda_{i,1} + \lambda_{i,2}}, \tag{4.22}$$

where $\lambda^*_{i,2}$ is the number of closed 4-paths, centred at node u_i, that form a 6-cycle with two chords and $\lambda_{i,2}$ the total number of 4-paths with two chords that include node u_i.

The three chord local clustering coefficient:

$$dcc_{i,3} = \frac{\lambda^*_{i,3}}{\lambda_{i,2}}, \tag{4.23}$$

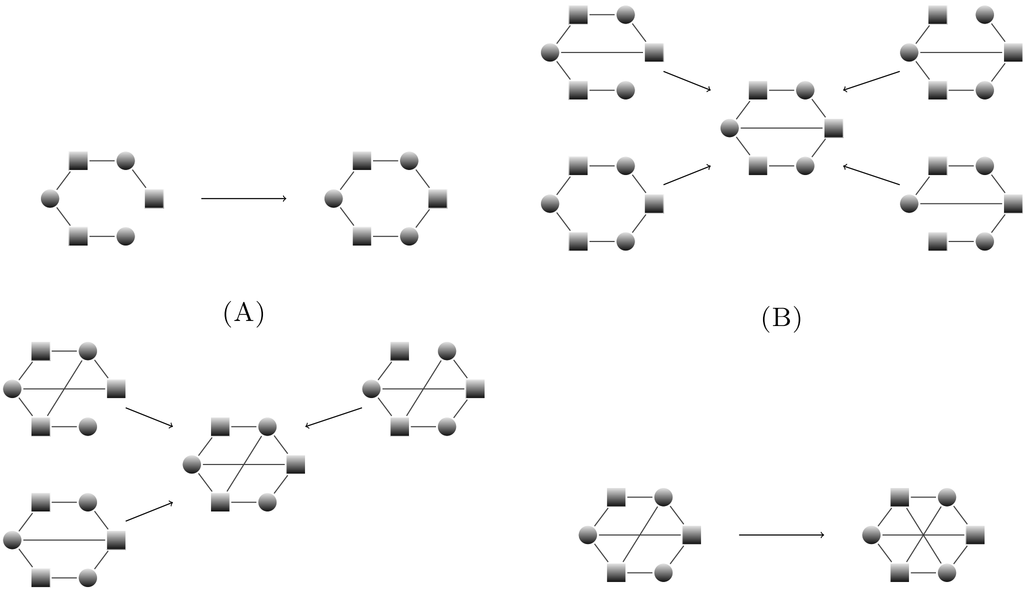
where $\lambda^*_{i,3}$ is the number of closed 4-paths, centred at node u_i, that form a 6-cycle with three chords.

Since a path of length four can start and end at a primary node or it can start and end at a secondary node, it is possible to calculate the time dependent clustering coefficients in terms of the primary or in terms of the secondary node set. Therefore, if measuring triadic closure between the nodes of the, say primary node set, one would consider all 4-paths that start and end at primary nodes.

4.6 A clustering coefficient for time independent networks

Similar to the time dependent clustering coefficient, we introduce four different clustering coefficients for time independent bipartite networks that correspond to the four structures depicted in Figure 4.8.

In a time independent bipartite network, cycles form differently to those formed in time dependent networks. In contrast to time dependent networks, primary nodes can form connections to secondary nodes at any point in time. Figure 4.10 illustrates the different possibilities by which the differently structured 6-cycles may be formed in a time independent network.



(A)

(B)

(C)

(D)

FIGURE 4.10: All possibilities by which the different 6-cycles may be formed in a time independent network. (A) We call the origin of an induced 6-cycle an induced 5-path. (B) An induced 6-cycle may form a 6-cycle with one chord at the next time step. We call a 5-path that contains an extra edge between two of its nodes that does not belong to the path, a 5-path with one chord. Any of the three different 5-paths with one chord may form a 6-cycle with one chord. (C) A 6-cycle with one chord may form a 6-cycle with two chords at the next time step. We call a 5-path that contains two extra edges between its nodes that do not belong to the path, a 5-path with two chords. Any of the two different 5-paths with two chords may form a 6-cycle with two chords. (D) A 6-cycle with two chords may form a 6-cycle with three chords at the next time step.

Equations (4.24) - (4.27) give four clustering coefficients icc_x, one for each type of 6-cycle, that suit the analysis of time independent bipartite networks, such as rating networks.

The induced clustering coefficient:

$$icc_0 = \frac{6\sigma_0}{6\sigma_0 + \kappa_0}, \tag{4.24}$$

where σ_0 is the number of induced 6-cycles and κ_0 the number of induced 5-paths. The induced clustering coefficient icc_0 measures the proportion of induced 6-cycles to all induced paths of length five (see Figure 4.10A). The number of induced 6-cycles is multiplied by six, as each induced 6-cycle contains six induced 5-paths.

The one chord clustering coefficient:

$$icc_1 = \frac{7\sigma_1}{7\sigma_1 + \sigma_0 + \kappa_1}, \tag{4.25}$$

where σ_1 is the number of induced 6-cycles with one chord and κ_1 the number of 5-paths with one chord. The one chord clustering coefficient icc_1 measures the proportion of 6-cycles with one chord with respect to its origins (see Figure 4.10B). The number of 6-cycles with one chord is multiplied by seven, as each contains two of each of the 5-paths with one chord, shown in Figure 4.10B, and one 6-cycle without any chords.

The two chord clustering coefficient:

$$icc_2 = \frac{4\sigma_2}{4\sigma_2 + \sigma_1 + \kappa_2}, \tag{4.26}$$

where σ_2 is the number of induced 6-cycles with two chords and κ_2 the number of 5-paths with two chords. The two chord clustering coefficient icc_2 measures the proportion of 6-cycles with two chords with respect to its origins (see Figure 4.10C). The number of 6-cycles with two chords is multiplied by four, as it contains one of each of the 5-paths with two chords, shown in Figure 4.10C, and two 6-cycles with one chord.

The three chord clustering coefficient:

$$icc_3 = \frac{3\sigma_3}{3\sigma_3 + \sigma_2}, \tag{4.27}$$

where σ_3 is the number of induced 6-cycles with three chords. The three chord clustering coefficient icc_3 measures the proportion of 6-cycles with respect to its origin (Figure 4.10D). The number of 6-cycles with three chords is multiplied by three, as each contains three 6-cycles with two chords.

In a one-mode network, the local clustering coefficient of a node u_i is calculated by dividing the number of closed 2-paths that are centred at node u_i by the total number of 2-paths that are centred at the node. In rating networks, most clusters are formed from 5-paths (see Figure 4.10). As a path of odd length can never be centred at a node, we consider all paths that involve node u_i in order to calculate u_i's clustering coefficient. The local clustering coefficients are denoted $icc_{i,x}$ and are given by the following four equations:

The induced local clustering coefficient:

$$icc_{i,0} = \frac{6\sigma_{i,0}}{6\sigma_{i,0} + \kappa_{i,0}}, \tag{4.28}$$

where $\sigma_{i,0}$ is the number of induced 6-cycles that include node u_i and $\kappa_{i,0}$ the number of induced 5-paths that include node u_i.

The one chord local clustering coefficient:

$$icc_{i,1} = \frac{7\sigma_{i,1}}{7\sigma_{i,1} + \sigma_{i,0} + \kappa_{i,1}}, \tag{4.29}$$

where $\sigma_{i,1}$ is the number of induced 6-cycles with one chord that include node u_i and κ_1 the number of 5-paths with one chord that include node u_i.

The two chord local clustering coefficient:

$$icc_{i,2} = \frac{4\sigma_{i,2}}{4\sigma_{i,2} + \sigma_{i,1} + \kappa_{i,2}}, \tag{4.30}$$

where $\sigma_{i,2}$ is the number of induced 6-cycles with two chords that include node u_i and $\kappa_{i,2}$ the number of 5-paths with two chords that include node u_i.

The three chord local clustering coefficient:

$$icc_{i,3} = \frac{3\sigma_3}{3\sigma_3 + \sigma_2}, \tag{4.31}$$

where σ_3 is the number of induced 6-cycles with three chords that include node u_i.

4.7 Summary

Many real world networks are bipartite. However, not every network measure can be directly applied to this type of network [61]. In order to analyse bipartite networks, one can either project the bipartite network onto a one-mode network or redefine network measures to suit the analysis of bipartite networks. We formally showed that the clustering coefficient of the projection of a random bipartite is higher than that of a random one-mode network of the same order, size and with identical degree sequence. Thus, comparing the projection of a bipartite network to random one-mode networks may lead to false conclusions. Although this is well known, ours is the first formal confirmation.

Network scientists have proposed clustering coefficients for bipartite networks, however, most do not consider triadic closure [70, 109, 134]. Opsahl [94] who has considered closure between three nodes, ignores the different structures that a bipartite cluster may have.

This chapter illustrated the importance of distinguishing between different types of 6-cycles that are identified by the number of chords connecting nodes within the cycle. Ignoring these chords results in an over-count of 6-cycles. We demonstrated that the formation of the different types of 6-cycles depends on the development of the network over time. For instance, in a time independent network where primary nodes may connect to secondary nodes at any point in time, a 6-cycle with exactly one chord could originate from an induced 6-cycle. This is not possible in a time dependent network where any type of 6-cycle always originates from a 4-path. We defined four clustering coefficients that correspond to the different types of 6-cycles for time dependent and time independent networks.

In the next chapter, we look at applications of the bipartite clustering coefficients.

Chapter 5

Applications of the Clustering Coefficient

Parts of this chapter have been published in [66] and [67].

5.1 Introduction

5.1.1 Motivation

The previous chapter was motivated by the inability to apply the one-mode clustering coefficient to bipartite networks. We dealt with this limitation by defining bipartite clustering coefficients that are suited for the analysis of different types of bipartite networks.

The one-mode clustering coefficient is an important measure mainly because of its many applications. Chen et al. [20] for instance, have shown that in a one-mode network a node with a low local clustering coefficient is able to spread a disease or information much quicker than a node with a high local clustering coefficient.

The question arises whether the bipartite clustering coefficients are also able to do this, particularly since many real world networks are bipartite [61].

This chapter explores the applications of the novel bipartite clustering coefficients that we introduced in Chapter 4 to real world datasets.

5.1.2 Outline

Our contributions in this chapter are the following: We use the bipartite clustering coefficients to detect the most important nodes within real world bipartite networks by introducing the concept of the driving score. We use the clustering coefficients to predict the popularity of new items in rating networks.

This chapter is divided into two parts: Section 5.2 examines the identification of influential nodes in complex networks. Section 5.3 investigates methods of predicting the popularity of new items in rating networks. Section 5.2 begins with a review of previous results in discovering influential nodes in one-mode networks. We then define a novel measure that combines our bipartite clustering coefficients (see Chapter 4) to measure the extent of influence for each node in a bipartite network. This measure is then applied to two real world networks. Section 5.3 starts by reviewing existing methods of predicting the popularity of items in online rating networks. Next we introduce a clear definition of the term *popularity* in the context of rating networks. We then utilise our bipartite clustering coefficient to separately predict the number of ratings and the average rating of new items in two real world networks. We conclude the chapter with a summary in Section 5.4

5.2 Identification of influential nodes

Locating influential nodes in a network is often crucial as this could aid in stopping the spread of diseases or alternatively assist the spread of knowledge and information [22]. Pastor-Satorras and Vespignani [98] have shown that the dynamics of large complex networks are often controlled by a small number of influential nodes. According to the Oxford dictionary the word *influence* is defined as follows:

The capacity to have an effect on the character, development, or behaviour of someone or something, or the effect itself.

Henceforth this definition of *influence* is used. From this definition it is clear that influential nodes would be important to the development and behaviour of the whole network. Consequently we use the terms *influential* and *important* interchangeably in the remainder of this thesis.

Against intuition, the most influential nodes are not necessarily those with the highest degrees [20, 22, 55, 135]. Although a node's degree centrality can be easily determined (see Subsection 2.2.3.2), it may not always reveal the most important and influential nodes in the network. Nodes with high degree could be located on the periphery of the network, thus failing to influence a large proportion of other nodes [55]. In addition, networks often exhibit community structure, with groups of nodes being clustered together, and it may be the case that all nodes with high degrees belong to a single community [135].

Other, more complex, centrality measures such as betweenness, given by Equation (2.3), and closeness, given by Equation (2.4), prove to be inefficient in finding the truly important nodes of a complex network [22, 55]. The frequent discussion of this problem in the literature has led to different approaches to finding influential nodes in one-mode networks [20–22, 55, 135].

5.2.1 Node location

Kitsak et al. [55] discovered that the location of a node within a network determines its influence. The authors argue that a node with low degree, holding a key location in the network core, can influence a higher number of nodes than a vertex with high degree that is located on the periphery of the network.

To find the network core, Kitsak et al. [55] use k-shell decomposition. Decomposing a network into shells leads to an allocation of nodes to different shells of the network. One starts by removing all nodes of degree one until all remaining nodes have degree two or higher. Note that nodes of degree greater that one may have degree one after removing all nodes that initially have degree one. These nodes are then also removed. The removed nodes form the k_1-shell. Next all nodes of degree two are removed and so on until every node is assigned to exactly one shell. As an example consider Figure 5.1.

The results presented in [55] provide evidence that a node located in the core has higher spreading ability than a node with high degree. Note here that as per Kitsak et al. [55] the network core is formed by nodes with high k-shell index.

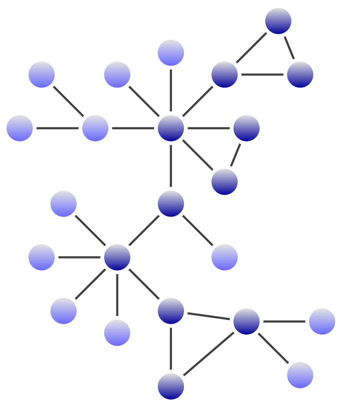


FIGURE 5.1: A small network that is decomposed into its shells. The light blue nodes belong to the k_1-shell, the dark blue nodes belong to the k_2-shell.

5.2.2 The role of clustering

Chen et al. [20] agree with Kitsak et al. [55] that the location of a node is more important than its degree, closeness centrality or betweenness centrality. The authors point out that a node with a few influential neighbours may be more important than a node with a large number of unimportant neighbours. Further, the density of connections between the neighbours of a node, measured by the local clustering coefficient, may impact its level of importance [20].

Hence, the authors propose an algorithm called *ClusterRank* that, based on the local one-mode clustering coefficient, orders the nodes of a network by importance. Tests show that ClusterRank performs better than common centrality measures, PageRank [95] and LeaderRank [71]. Note that in an undirected network PageRank and LeaderRank are both equivalent to degree centrality [20]. The results presented in [20] indicate that a high local clustering coefficient hinders a node from spreading information or a disease rapidly throughout the network.

According to the research carried out on one-mode networks, a node's influence level is not determined by its degree. However, the location of a node within a network and the connectedness of its neighbourhood are important factors that impact importance. Considering local measures such as the individual clustering behaviour of nodes, rather than global measures, ensures that influential nodes are identified across different network communities [135].

Since the local clustering coefficient of a node in a one-mode network is a promising indicator of its importance, we apply the bipartite clustering coefficients that we developed in Chapter 4 to identify influential nodes in real world bipartite networks.

In contrast to the one-mode clustering coefficient, we have four bipartite clustering coefficients (Equations (4.16) - (4.19) for time dependent networks, and Equations (4.24) - (4.27) for time independent networks) corresponding to the different 6-cycles depicted in Figure 4.8. In what follows, we suggest a way of combining the four clustering coefficients, ultimately leading to the detection of influential nodes.

5.2.3 The driving score

We now define a novel measure, called the driving score, that assigns a value to a bipartite network and to each node in the network, reflecting the extent to which each node is driving the whole network away from being random. Our idea behind the driving score is to determine each node's contribution to the global clustering behaviour of the complete network. As it is not clear if a low local bipartite clustering coefficient implies a high level of influence, as is often the case in one-mode networks, we compare the bipartite network of interest to random networks. Although Chen et al. [20] found that high local clustering hinders information spread, this may not be the case in a bipartite network, as the first neighbours of any node cannot be connected to each other.

In the following we describe the calculation of driving scores for time dependent bipartite networks. The driving scores for time independent networks are calculated in the exact same manner.

Firstly, all four global clustering coefficients of the network are calculated and compared to the clustering coefficients of an ensemble of similar random networks (networks of the same order and size, and having identical degree sequences). There are two cases:

- $dcc_x < \mu_x$ or
- $dcc_x \geq \mu_x$,

where μ_x is the mid point of the 95% confidence interval, ie. the mean, of dcc_x, calculated in the ensemble of random networks. We define the global driving score, denoted ds, of a network to be the normalised average distance between the global clustering coefficients dcc_x and the mean μ_x in the ensemble of random networks:

$$ds = \begin{cases} \frac{1}{4}\sum_{x=0}^{3}(|\mu_x - dcc_x|)/\mu_x & \text{if} \quad dcc_x < \mu_x, \\ \frac{1}{4}\sum_{x=0}^{3}(|\mu_x - dcc_x|)/(1-\mu_x) & \text{if} \quad dcc_x \geq \mu_x. \end{cases} \tag{5.1}$$

We consider the four different 6-cycles, depicted in Figure 4.8, equally important, hence the factor of 1/4 in Equations (5.1) and (5.2).

Note that ds ranges between zero and one. The greater the average difference between the global clustering coefficients dcc_x and the respective mean μ_x, the higher the global driving score.

We now determine the driving scores of each individual node by comparing its local clustering coefficients to the confidence intervals of the global clustering coefficients. A node u_i with a local clustering coefficient $dcc_{i,x}$ that is close to μ_x behaves as expected and hence does not contribute to a global clustering behaviour that is different from a random network. Given the global clustering coefficient dcc_x of a network is smaller than μ_x, i.e. $dcc_x < \mu_x$, then either

- $dcc_{i,x} < \mu_x$ or
- $dcc_{i,x} \geq \mu_x$.

If the local clustering coefficient $dcc_{i,x}$ of node u_i also lies below μ_x, then node u_i contributes to the global clustering behaviour of the whole network and we assign a score in the interval $[0, 1]$ to node u_i, depending on the difference between the local clustering coefficient and the mean μ_x in the ensemble of random networks. If on the other hand, $dcc_{i,x}$ lies above μ_x then node u_i drives against the clustering behaviour and we assign a score in the interval $[-1, 0]$ to node u_i. Similarly, there are two cases when the global clustering coefficient dcc_x lies above the mid point of the confidence interval.

The local driving score ds_i of node u_i is thus given by the following equation:

$$ds_i = \begin{cases} \frac{1}{4}\sum\limits_{x=0}^{3}(|\mu_x - dcc_{i,x}|)/\mu_x & \text{if} \quad dcc_x < \mu_x > dcc_{i,x}, \\ -\frac{1}{4}\sum\limits_{x=0}^{3}(|\mu_x - dcc_{i,x}|)/(1-\mu_x) & \text{if} \quad dcc_x < \mu_x \leq dcc_{i,x}, \\ \frac{1}{4}\sum\limits_{x=0}^{3}(|\mu_x - dcc_{i,x}|)/(1-\mu_x) & \text{if} \quad dcc_x \geq \mu_x \leq dcc_{i,x}, \\ -\frac{1}{4}\sum\limits_{x=0}^{3}(|\mu_x - dcc_{i,x}|)/\mu_x & \text{if} \quad dcc_x \geq \mu_x > dcc_{i,x}. \end{cases} \tag{5.2}$$

Figure 5.2 illustrates that for a node to achieve a high driving score, it does not necessarily have to have high clustering coefficients. For instance, if the global clustering coefficients are lower than their respective confidence intervals, only a node with low clustering coefficients can receive a high driving score.

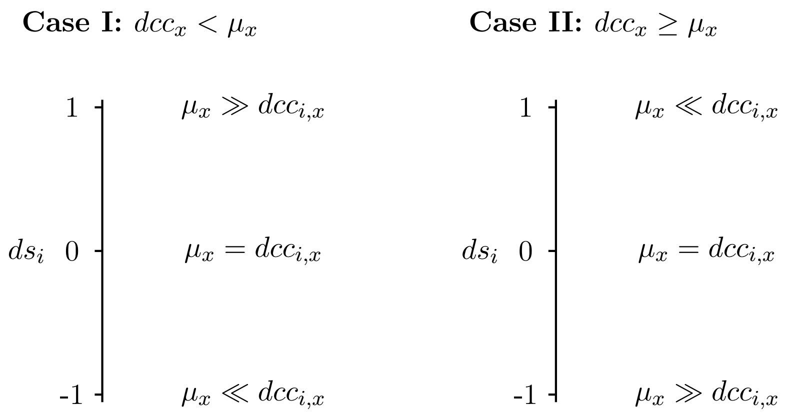


FIGURE 5.2: If $dcc_x < \mu_x$ node u_i receives a high driving score only if $dcc_{i,x} \ll \mu_x$. Similarly, if $dcc_x \geq \mu_x$ node u_i receives a high driving score only if $dcc_{i,x} \gg \mu_x$.

We can now apply the driving score technique to find influential nodes in real world networks.

5.2.4 The Southern Women network

The Southern Women network consists of 18 women and 14 events [27]. An edge between a woman and an event indicates that the woman has attended the event. More detail on the dataset is provided in Chapter 2 (see Subsection 2.4.7). This bipartite network is clearly time dependent as the events take place at a certain time and cannot be attended afterwards.

Table 5.1 shows the clustering coefficients of the Southern Women network and the average coefficients of an ensemble of 100 random bipartite networks together with the 95% confidence intervals with respect to the set of women. The random networks were created by applying the Curveball algorithm (see Subsection 2.2.5.2) to the Southern Women network. Hence, the random networks have the same order and size as well as the same degree sequences as the Southern Women network.

	Southern Women network	Average random network
dcc_0	0.4446	0.6370 [0.6261, 0.6478]
dcc_1	0.6532	0.5571 [0.5483, 0.5658]
dcc_2	0.5984	0.4105 [0.3972, 0.4237]
dcc_3	0.5604	0.3238 [0.3018, 0.3457]

TABLE 5.1: The four global clustering coefficients of the Southern Women network with respect to the set of women and the average clustering coefficients of 100 randomly generated networks with their 95% confidence intervals.

None of the four global clustering coefficients of the Southern Women network lie within the 95% confidence intervals and hence, none of the values are as expected in a similar random network. The coefficient dcc_0 lies below the lower bound of the confidence interval whereas dcc_1, dcc_2 and dcc_3 lie above the interval. In the average random network dcc_0 has the highest value ($dcc_0 = 0.6370$), as opposed to the Southern Women network, where dcc_0 takes the lowest value ($dcc_0 = 0.4446$). Hence, a greater proportion of 4-paths are closed to form an induced 6-cycle in a random network than in the Southern Women network. The remaining three clustering coefficients lie above the 95% confidence interval, showing that the proportion of closed 4-paths with chords is much higher than expected. Since the Southern Women network is a social network, one would assume that any of the 18 women would rather attend an event with friends than by herself. Our results confirm the assumption that three women tend to cluster if they are already connected to each other by at least one event.

5.2.4.1 Ranking by driving score

The global driving score with respect to the women of the Southern Women network is $ds = 0.297$. The local clustering coefficients of the individual women, together with their respective driving scores are displayed in Table 5.2. A positive driving score indicates that the local clustering coefficients are contributing to the global clustering behaviour of the whole network. A negative driving score indicates that the clustering behaviour of

Woman i	$dcc_{i,0}$	$dcc_{i,1}$	$dcc_{i,2}$	$dcc_{i,3}$	ds_i
Evelyn	0.3957 ↓	0.6986 ↑	0.6732 ↑	0.6545 ↑	0.4083
Laura	0.4468 ↓	0.6610 ↑	0.7218 ↑	0.7364 ↑	0.4179
Theresa	0.0619 ↓	0.7228 ↑	0.7951 ↑	0.6667 ↑	0.6092
Brenda	0.3455 ↓	0.656 ↑	0.7241 ↑	0.7565 ↑	0.4633
Charlotte	1 ↑	0.84 ↑	0.6093 ↑	0.6 ↑	0.0962
Frances	0.6667 ↑	0.684 ↑	0.5164 ↑	0.7742 ↑	0.2626
Eleanor	0.5094 ↓	0.662 ↑	0.6302 ↑	0.6234 ↑	0.3133
Pearl	0.4074 ↓	0.6931 ↑	0.4278 =	0.0652 ↓	-0.0254
Ruth	0.2869 ↓	0.697 ↑	0.6254 ↑	0.3704 =	0.3248
Verne	0.3778 ↓	0.613 ↑	0.6188 ↑	0.3429 =	0.2253
Myrna	0.6735 ↑	0.5221 ↓	0.504 ↑	0.4615 ↑	0.0498
Katherine	0.7260 ↑	0.569 ↑	0.5572 ↑	0.5254 ↑	0.0822
Sylvia	0.3395 ↓	0.6694 ↑	0.653 ↑	0.5444 ↑	0.3646
Nora	0.7185 ↑	0.7555 ↑	0.4021 ↓	0.5238 ↑	0.1247
Helen	0.7143 ↑	0.6273 ↑	0.4703 ↑	0.375 =	0.0308
Dorothy	0.4667 ↓	0.4557 ↓	0.163 ↓	0 ↓	-0.3793
Olivia	1 ↑	0.3103 ↓	0 ↓	0 ↓	-0.8607
Flora	1 ↑	0.3103 ↓	0 ↓	0 ↓	-0.8607

TABLE 5.2: The local clustering coefficients and driving scores of the 18 women. We identified seven women to be driving the global clustering behaviour (women with a higher driving score than the global driving score) - these women's names are printed in bold. The arrows next to the entries indicate whether the local clustering coefficient lies above (↑) or below (↓) the confidence interval of the respective global clustering coefficient.

the node drives against the global clustering behaviour. The arrows next to the entries in Table 5.2, indicate whether the local clustering coefficient lies above (↑) or below (↓) the confidence interval of the respective global clustering coefficient.

	Southern Women network	Average random network
dcc_0	0.3578	0.7288 [0.7164, 0.7412]
dcc_1	0.597	0.6386 [0.6272, 0.6500]
dcc_2	0.8556	0.5040 [0.4871, 0.5209]
dcc_3	0.7903	0.4489 [0.4220, 0.4757]

TABLE 5.3: The four global clustering coefficients of the Southern Women network with respect to the secondary node set of events and the average clustering coefficients of 100 randomly generated networks with the 95% confidence interval.

The driving scores of the women reveal that seven women (Evelyn, Laura, Theresa, Brenda, Eleanor, Ruth and Sylvia) heavily influence the clustering behaviour of the whole network. Their driving scores lie above the global driving score of the network and hence we consider these women to be the most influential as they are driving the total network away from being random. All nodes with a negative score drive against the overall clustering behaviour. These individuals fail to change the global clustering behaviour and hence are considered not influential. The women with a negative driving score are Pearl, Dorothy, Olivia and Flora.

We repeated the analysis for the secondary node set that represents the 14 events. Table 5.3 shows the global clustering coefficients of the Southern Women network with

Event i	$dcc_{i,0}$	$dcc_{i,1}$	$dcc_{i,2}$	$dcc_{i,3}$	ds_i
1	1 $\uparrow$	0.9556 $\uparrow$	0.7714 $\downarrow$	0.6 $\downarrow$	-0.2659
2	0.8 $\uparrow$	0.9574 $\uparrow$	0.8571 $\uparrow$	0.5143 $\downarrow$	-0.0785
3	0.3043 $\downarrow$	0.7113 $\uparrow$	0.9727 $\uparrow$	0.8824 $\uparrow$	0.5281
4	0.9 $\uparrow$	0.9529 $\uparrow$	0.8803 $\uparrow$	0.6427 $\uparrow$	-0.0976
5	0.2545 $\downarrow$	0.7952 $\uparrow$	0.9895 $\uparrow$	0.9029 $\uparrow$	0.5050
6	0.3421 $\downarrow$	0.5482 $\downarrow$	0.8913 $\uparrow$	0.8791 $\uparrow$	0.5584
7	0.3195 $\downarrow$	0.6965 $\uparrow$	0.8165 $\uparrow$	0.7051 $\uparrow$	0.3740
8	0.38 $\downarrow$	0.5918 $\downarrow$	0.9429 $\uparrow$	0.8672 $\uparrow$	0.5489
9	0.3062 $\downarrow$	0.6823 $\uparrow$	0.7968 $\uparrow$	0.6923 $\uparrow$	0.3727
10	0.48 $\downarrow$	0.7023 $\uparrow$	0.7891 $\uparrow$	0.8049 $\uparrow$	0.3465
11	1 $\uparrow$	0.7949 $\uparrow$	0.1 $\downarrow$	0 $\downarrow$	-0.8085
12	0.3889 $\downarrow$	0.7348 $\uparrow$	0.8187 $\uparrow$	0.875 $\uparrow$	0.4019
13	1 $\uparrow$	0.6098 $\downarrow$	0.5323 $\uparrow$	0.6923 $\uparrow$	-0.1140
14	1 $\uparrow$	0.6098 $\downarrow$	0.5323 $\uparrow$	0.6923 $\uparrow$	-0.1140

Table 5.4: The local clustering coefficients and the driving scores of the 14 events. The events that we identified as driving the global clustering behaviour are printed in bold. The arrows next to the entries indicate whether the local clustering coefficient lies above ($\uparrow$) or below ($\downarrow$) the confidence interval of the respective global clustering coefficient.

respect to the events. We remind the reader that in a time dependent bipartite network the global clustering coefficients with respect to the primary node set are not equal to the global clustering coefficients with respect to the secondary node set (see Section 4.5). The global driving score with respect to the events equals 0.4756.

Again, none of the four clustering coefficients lie within the 95% confidence interval. The coefficients dcc_0 and dcc_1 lie below the lower bound of the respective confidence interval whereas the dcc_2 and dcc_3 lie above the intervals.

Calculating the driving scores of the 14 events, listed in Table 5.4, shows that events 3, 5, 6 and 8 drive the clustering behaviour of the network.

5.2.4.2 Discussion

The first analysis of the Southern Women dataset, carried out by Davis et al. [27] in the form of interviews with the aim of categorising the 18 women into groups, found two different groups that were further divided into core, primary and secondary members. Figure 5.3 shows the Southern Women network with the two groups of women as identified in [27]. Our analysis found that all four core women of the first group are influential as well as one core woman of the second group. Interestingly, our results show that Eleanor and Ruth are also influential. Both attended only four events, however, these events were also attended by members from both groups. This observation indicates that Eleanor and Ruth are important connections between the two groups. Davis et al.

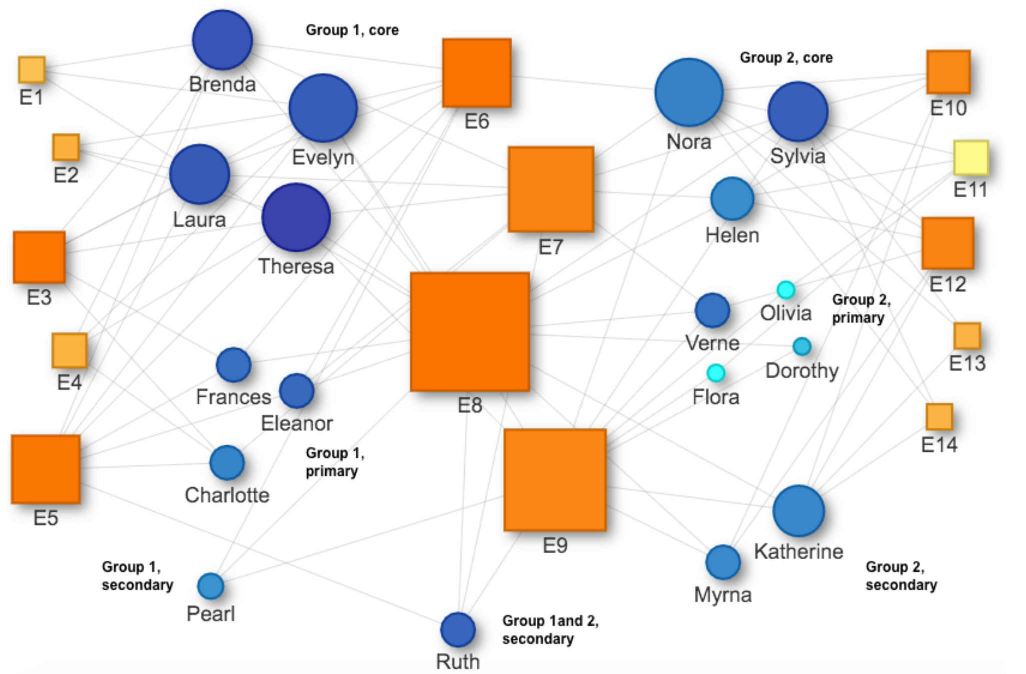


FIGURE 5.3: The Southern Women network with the two groups of women as identified in [27]. The size of the nodes corresponds to their degrees. The darker the shading of the node, the higher its driving score.

[27] also found that Ruth had some affiliation with both groups. Clearly, our clustering coefficient identifies important nodes across the communities that were identified in [27]. Our analysis shows that the importance of a woman within the network does not depend on her degree. For instance, if Ruth, who has a low degree, is removed from the network, information would spread less easily between the two groups.

Dorothy, Olivia, Flora and Pearl received negative driving scores and hence are not considered influential. While driving against the global clustering behaviour, they fail to change the global clustering behaviour of the network. Although all four women were associated with a group by Davis et al. [27], the results presented in [13, 29, 34] show no group association for Dorothy, Olivia, Flora and Pearl.

The important events, identified by our analysis, seem to form connections between the two groups as well as between core, primary and secondary members of the individual groups. Unfortunately, we do not have any information about the nature of the events to be able to provide any insight as to why this may be the case.

Literature on network science agrees that degree, betweenness and other centrality measures generally fail to identify the truly influential nodes of a network. There are numerous examples of particular situations, for instance when the network has a community structure. Our method can be applied in the general case when simple measures fail as well as in the rare case when they perform well.

Our analysis of this dataset confirms this, showing that the importance of a woman does not depend on her degree.

5.2.5 The Noordin Top terrorist network

In this subsection we look at a subset of the Noordin Top terrorist network [35] that contains 26 of the 79 known members of the terrorist ring (see Chapter 2, Subsection 2.4.6 for more details on this dataset). This sub-network models the attendance of these 26 members at 20 different meetings. It contains a total of 64 connections between members and meetings. Table 5.5 shows the global clustering coefficients of the network with respect to the members of the terrorist ring.

	Noordin Top Terrorist Network	Average random network
dcc_0	0.0303	0.1871 [0.1768, 0.1973]
dcc_1	0.1108	0.0609 [0.0542, 0.0676]
dcc_2	0.2	0.0288 [0.0199, 0.0376]
dcc_3	0	0.0074 [0, 0.0148]

TABLE 5.5: The four global clustering coefficients of the terrorist network and the average global clustering coefficients of 100 randomly generated networks with the 95% confidence interval with respect to the members of the terrorist ring.

The clustering coefficient dcc_0 lies below the confidence interval, whereas dcc_1 and cc_2 lie above the interval. The clustering coefficient dcc_3, however, lies within the 95% confidence interval.

In the terrorist network, the proportion of 4-paths that are closed and form a 6-cycle with two chords is much higher than in a random network, giving $dcc_2 = 0.2$. As in the Southern Women network, it seems that three members of the terrorist ring would cluster if they were already connected through at least one previous meeting. The results from the analysis of the Southern Women network can be explained by the underlying friendship network. In case of the terrorist network, it is rather unlikely that the members themselves decided which meetings to attend, based on personal relationships to other members. Again, there is insufficient information about the terrorist ring available to explain this observation.

Member i	$dcc_{i,0}$	$dcc_{i,1}$	$dcc_{i,2}$	$dcc_{i,3}$	ds_i
Abdullah Sunata	n/a	n/a	n/a	n/a	n/a
Abu Dujanah	n/a	0 $\downarrow$	0.1667 $\uparrow$	0 =	0.0473
Abu Fida	0.1667 $\downarrow$	0.1333 $\uparrow$	0 $\downarrow$	n/a	-0.2713
Adung	n/a	n/a	n/a	n/a	n/a
Ahmad Rofiq Ridho	0.0408 $\downarrow$	0.1818 $\uparrow$	0.2414 $\uparrow$	0 =	0.5324
Akram	n/a	n/a	n/a	n/a	n/a
Asep Jaja	n/a	n/a	n/a	n/a	n/a
Azhari Husin	0 $\downarrow$	0.0842 $\uparrow$	0.2857 $\uparrow$	0 =	0.5723
Cholily	n/a	n/a	n/a	n/a	n/a
Heri Sigu Samboja	n/a	n/a	n/a	n/a	n/a
Imam Bukhori	n/a	n/a	n/a	n/a	n/a
Ismail	n/a	n/a	n/a	n/a	n/a
Iwan Dharmawan	0.1429 $\downarrow$	0.2609 $\uparrow$	0 $\downarrow$	n/a	-0.1836
Jabir	n/a	n/a	n/a	n/a	n/a
Joko Triharmanto	n/a	n/a	n/a	n/a	n/a
Misno	n/a	n/a	n/a	n/a	n/a
Mohamed Saifuddin	0 $\downarrow$	0 $\downarrow$	n/a	n/a	0
Noordin Mohammed Top	0.0141 $\downarrow$	0.124 $\uparrow$	0.2079 $\uparrow$	0 =	0.5442
Purnama Putra	0.1429 $\downarrow$	0.3333 $\uparrow$	0.3333 $\uparrow$	0 =	0.46
Qotadah	n/a	0 $\downarrow$	0.1667 $\uparrow$	0 =	0.0473
Saptono	n/a	n/a	n/a	n/a	n/a
Son Hadi	0.1667 $\downarrow$	0 $\downarrow$	n/a	n/a	-0.4454
Suramto	n/a	n/a	n/a	n/a	n/a
Ubeid	n/a	n/a	n/a	n/a	n/a
Urwah	0.2 =	0.3333 $\uparrow$	0 $\downarrow$	n/a	-0.2419
Usman bin Sef	0 $\downarrow$	0 $\downarrow$	n/a	n/a	0

TABLE 5.6: The table shows the local clustering coefficients and driving scores of the 26 members of the Noordin Top terrorist network. The four members that we identified as driving the global clustering behaviour are printed in bold. The global driving score with respect to the members equals 0.2692. The entry n/a indicates that the local clustering coefficient of that member is undefined, ie. the number of 4-paths centred at this member is equal to zero. The arrows next to the entries indicate whether the local clustering coefficient lies above ($\uparrow$) or below ($\downarrow$) the confidence interval of the respective global clustering coefficient.

5.2.5.1 Ranking by driving score

Table 5.6 shows the ranking of the 26 members using the driving score technique. The global driving score with respect to the members equals 0.2692. The members who are driving the clustering behaviour of the network are Ahmad Rofiq Ridho, Azhari Husin and Noordin Mohammed Top.

5.2.5.2 Discussion

Two of the driving nodes, Noordin Mohammed Top and Azhari Husin, worked together to plan terrorist attacks, with Noordin Mohammed Top financing the attacks and Azhari Husin being in charge of building the bombs [10]. Ahmad Rofiq Ridho acted as a communicator between the members [35]. Purnama Putra also received a high driving score.

Like Ahmad Rofiq Ridho, he acted as a communicator between the members of the terrorist ring.

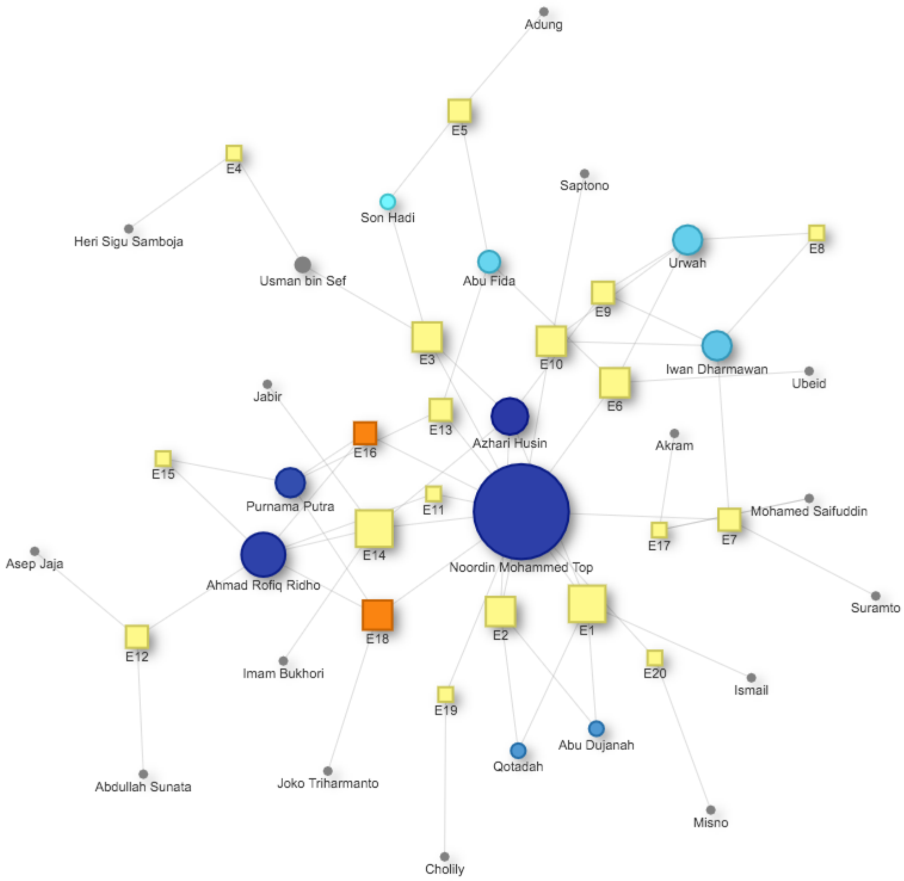


FIGURE 5.4: The Noordin Top terrorist network. The size of the nodes corresponds to their degrees. The darker the shading of the node, the higher its driving score.

The driving scores of the secondary node set revealed that meetings 16 and 18 are driving the clustering behaviour. Unfortunately, we do not have enough information about the meetings or the terrorist ring itself to explain these results.

Figure 5.4 depicts the Noordin Top terrorist network. The influential nodes, identified by the driving score technique, have a darker shading. Although Noordin Top is one of the most influential members in the network and also has the highest degree, in general the importance of a member is weakly correlated to its degree. The correlation coefficient between degrees and driving scores is 0.6.

This section demonstrated that the driving scores of the individual nodes in a network reveal those nodes that drive the global clustering behaviour and hence identify the most influential nodes in the network. Previous analyses support the results we obtained in this section [13, 27, 29, 34, 35].

5.3 Prediction of item popularity in rating networks

Many websites have the option to rate and review particular products. The information submitted by users is used to determine and predict current and future popularity of items. Prediction of the future popularity of new items that appear in a rating network, for example products on Amazon (`https://www.amazon.com/`), is challenging due to the lack of information. As the previous section demonstrated, the local clustering behaviour and hence the immediate neighbourhood of a node reveals much about its influence. The remainder of this chapter applies the bipartite clustering coefficients to the immediate neighbourhood of a new item in a rating network showing that this leads to good predictions of the item's future popularity.

Websites like Amazon, TripAdvisor and MovieLens offer their users a means to rate a variety of different items. Users can decide whether they are interested in an item based on its previously received ratings. Websites collect these user ratings for many reasons including recommending items to their users and predicting future item ratings [106]. The latter is rather challenging. In particular, new items that have received very few ratings to date are hard to classify as being popular or unpopular in the future, due to the sparsity of information [112]. It has been suggested that a ranking of the users may aid in improving predictions of future item popularity [133]. In other words, the behaviour of some users may be adopted by others. Here we examine the immediate neighbourhood of a new item, that is the clustering behaviour of the user who is the first to rate the new item, with the aim of predicting its future popularity. We demonstrate our approach on the MovieLens network [47] that contains ratings of 10,681 movies by 71,567 different users and the Digg network [49] that contains ratings of 3,553 stories by 139,409 users.

Zeng et al. [133] predict the popularity of items in three different bipartite networks, MovieLens, Digg and Netflix. Unfortunately, the Netflix dataset is no longer publicly available, so we are unable to compare our predictions to [133] for this particular dataset.

Zeng et al. [133] define popularity by the increase in degree, meaning that an item with a high increase in ratings is considered as popular. It is reasonable to assume that popular items are rated more frequently, but there may be exceptions. Some items may receive a relatively high number of low ratings and consequently should not be considered as popular. To improve predictions the authors also consider user influence, where a user is considered to be influential if he shows high rating activity. In one-mode networks, a high degree does not necessarily imply high influence [65]. Our work presented in the previous section also confirms that node degree is not a good indicator of influence. Besides giving a large number of ratings, an influential user should also give a wide range of ratings. A user who frequently rates items may not take enough time to thoroughly review the individual items. On the other hand, a user who rates items less frequently may give more reliable ratings.

Interestingly, although the approach by Zeng et al. [133] works well for items that have been in the network for some time, with a success rate of 72% for MovieLens, the fraction of new items correctly identified as popular in the future is small, approximately 30%. The success rate for new items in the Digg network was 20% and could be increased to 60% by using the friendship network of the Digg users that is also available. Since for most bipartite rating networks, the underlying friendship network of users is not available we did not consider it in this study.

Our main focus is to improve predictions for new items. To do so, we propose a more sophisticated method, utilising the bipartite clustering coefficients that we developed in Chapter 4. Our approach only considers network topology and is suitable for datasets such as MovieLens, that do not record any particulars of the users, such as age or gender. For our method to work, the knowledge of single ratings, whether high or low, is also unnecessary.

We demonstrate our method on two datasets, the MovieLens network and the Digg network. The MovieLens data [47] was collected by the University of Minnesota and contains 10,000,054 movie ratings that range between 1 and 5, with 5 being the best possible rating. Starting in January 1995, 71,567 different users rated 10,681 movies over a period of 14 years. Every user has a unique id but no additional information about the users is known. We formed a network by taking the users as the primary nodes and the movies as the secondary nodes. Each rating of a movie by a user is represented by an edge that links the user to the movie. Every edge is associated with a time stamp that corresponds to the time the rating was made.

Digg (`http://digg.com/`) is a website that features news stories and allows users to vote for them. This bipartite network contains 3,018,197 votes cast by 139,409 users [49]. A total of 3,553 stories were rated over a period of one month in 2009. Unlike the MovieLens network, edges are not associated with a rating. An edge between a user and a story indicates that the user liked the story. The data was obtained from `http://konect.uni-koblenz.de/networks/digg-votes`.

5.3.1 Defining popularity

Before making predictions about an item's future popularity, it is important to clearly define the meaning of popular and unpopular.

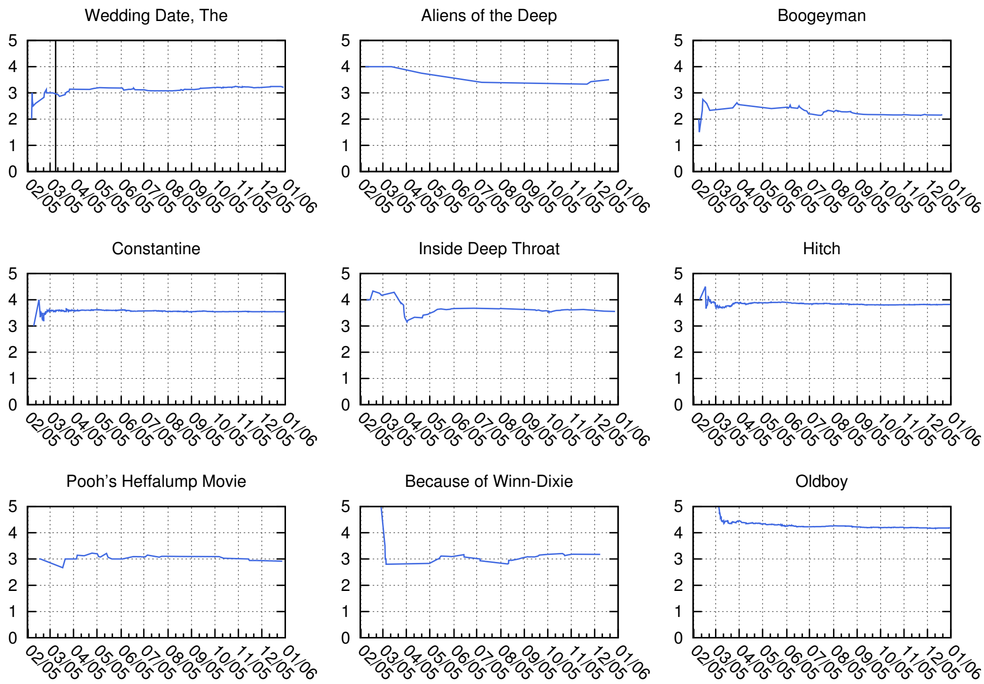


FIGURE 5.5: It is often the case that the average rating of a movie fluctuates during the first month after the initial rating, thereafter becoming steady. Here, we have plotted nine examples. The x-axis shows the time in form of the date and the y-axis displays the corresponding average rating.

In [133], an item is considered popular if its degree increased rapidly over a certain period of time. Often, a popular movie is watched and rated more often than an unpopular movie, however, there are exceptions. As mentioned before, a movie that receives a relatively high number of low ratings should certainly not be considered popular.

In contrast to the preferential attachment model [6, 102] that predicts that nodes with a high degree are much likelier to increase their degree than nodes with a low degree, in rating networks it is generally the case that the interest in an item, such as a movie, decays over time [74]. Although movies are rated over a longer period of time, the MovieLens data shows that in many cases the ratings made within one month of the movie's release determine its final average rating (see Figure 5.5). In the Digg dataset on the other hand, a new story is frequently rated within the first 48 hours. After this period the interest in the item decays rapidly (see Figure 5.6). This emphasises the need for good early predictions.

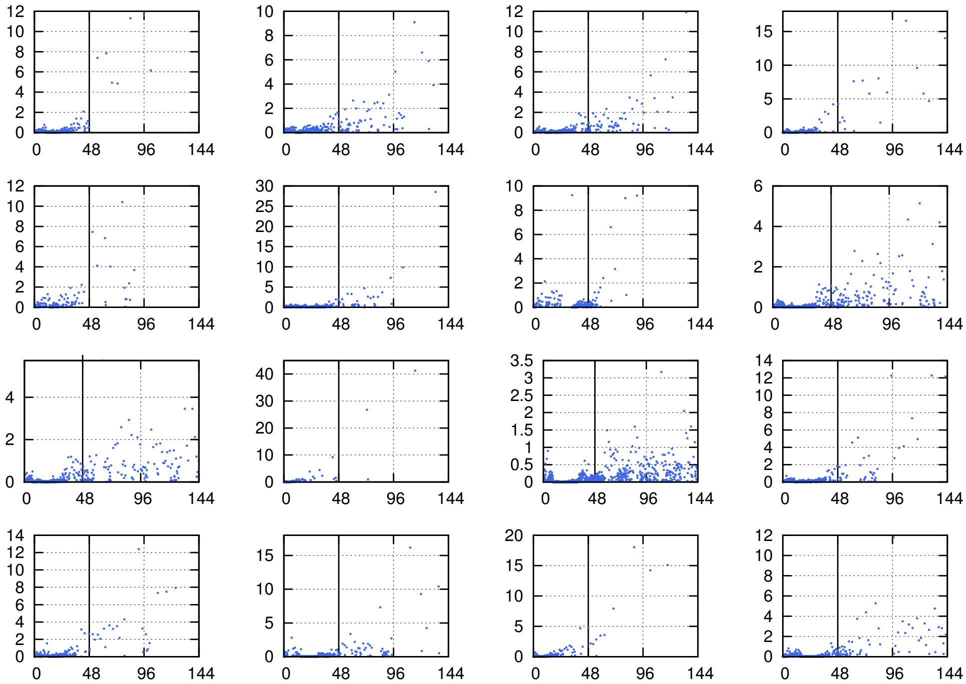


FIGURE 5.6: In the Digg network, interest in an item decays very quickly. Here we plotted the differences in time between the ratings of 16 different stories. The y-axis shows the difference in hours to the previous rating and the x-axis shows the time in hours after the first rating. Most items are frequently rated during the first 48 hours after the initial rating.

We define the critical period as the time period that most affects the average rating of an item. An item is considered as popular if it receives a higher number of ratings than the average item and simultaneously obtains a high average rating. In the MovieLens dataset, a movie received on average 29 ratings during the critical period (the first month after the movie's release). Hence, we consider a movie as popular if it receives 29 ratings or more during the first month and obtains an average rating greater or equal to 4. In the case of the Digg network, where edges are not associated with a rating, the critical period is the time span in which stories are most frequently rated, ie. within 48 hours of the initial rating.

We calculate a popularity score, denoted ρ, for each item based on the number of ratings received during the critical period and the average rating at the end of the critical period. In order to achieve a score that lies in the interval $[0, 1]$, a logistic function is used. A logistic function is an s-shaped curve that is frequently used to model population growth. The function grows exponentially at first with the slope decreasing thereafter until the function reaches a steady state. We define the popularity score ρ as:

$$\rho(\mu, n) = \frac{1}{1 + e^{-k(\mu n - c)}}, \tag{5.3}$$

where μ is the average rating, n is the number of ratings received within the critical period and c and k are constants. In the MovieLens dataset for instance, a movie is considered as popular if it received 29 ratings or more and obtained an average score of 4 or higher. The constant c is chosen such that a movie that receives exactly 29 ratings and an average score of 4 after the first month receives a popularity score of $\rho = 0.5$. The constant k is chosen such that an item without any ratings receives a popularity score of approximately zero. Note that $\rho = 0$ is undefined.

Hence, in case of the MovieLens network:

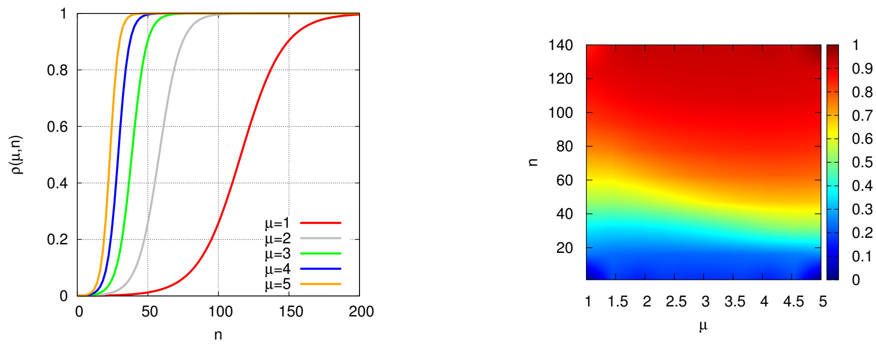
$$\begin{aligned} 0.5 &= 1/\left(1 + e^{-k(4 \cdot 29 - c)}\right) \\ 0.5(1 + e^{-k(116-c)}) &= 1 \\ e^{-k(116-c)} &= 1 \\ -k(116 - c) &= 0 \\ c &= 116 \quad \text{since} \quad k \neq 0 \end{aligned} \tag{5.4}$$

and

$$\begin{aligned}
0.5 \cdot 10^{-3} &= 1/\left(1 + e^{-k(0-116)}\right) \\
0.5 \cdot 10^{-3}(1 + e^{116k}) &= 1 \\
e^{116k} &= 1,999 \\
116k &= 7.6 \\
k &= 0.066.
\end{aligned} \tag{5.5}$$

Therefore $\rho(\mu, n) = 1/\left(1 + e^{-0.066(\mu n - 116)}\right)$ in the MovieLens network.

In the Digg network a story received on average six votes during the first two days. Digg does not give its users the opportunity to rate an item on a scale and an edge between a user and a story indicates that the user liked the story. To be able to calculate the popularity score ρ for items in such networks, we assign a rating of five to each edge.



(A) (B)

FIGURE 5.7: The popularity function ρ with regards to the MovieLens dataset. (A) We plotted Equation (5.3) for five different values of μ, in order to show the corresponding change in the shape of the logistic curve. Clearly, for a low average rating μ a large number of ratings is necessary to achieve a high popularity score. (B) The heat map shows values of the popularity score ρ with respect to the number of ratings and the average rating for the MovieLens network. Red corresponds to $\rho = 1$ and blue to $\rho \approx 0$. The heat map shows that a high value of ρ can only be achieved if the number of ratings as well as the average rating is high. The upper left corner of the heat map is also dark red, however, an item with a low average rating usually does not gain the requisite large number of ratings in order to receive a high popularity score.

Thus, a news story that received six ratings and an average rating of five should receive a popularity score of 0.5. Hence, $c = 30$, $k = 0.253$ and therefore $\rho(\mu, n) = 1/\left(1 + e^{-0.253(\mu n - 30)}\right)$ for the Digg network.

Figure 5.7A shows the change in the shape of $\rho(\mu, n)$ as the average rating μ changes in the case of the MovieLens dataset. Figure 5.7B shows a plot of $\rho(\mu, n)$ in the form of a heat map to give an alternative visualisation.

To demonstrate that the average rating of an item is not correlated to the number of ratings it received, we plotted the average ratings of movies against their degree (see Figure 5.8). If we were to consider only the number of received ratings to determine the movie's popularity, many would be wrongly classified as popular, see lower right quadrant of the plot. All movies in the lower right quadrant received a relatively high number of ratings but have an average rating of less than four.

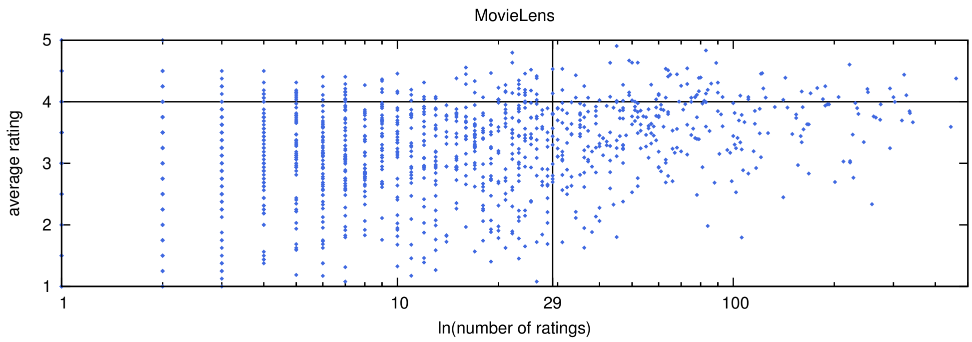


FIGURE 5.8: We plotted the movies' average ratings μ against the number of ratings n they received, to examine their relationship. The correlation coefficient between the average rating and the number of ratings is low (0.183). If we were to consider only the number of received ratings to determine the movie's popularity, many would be wrongly classified as popular, see lower right quadrant of the plot. All movies in the lower right quadrant received a relatively high number of ratings but have an average rating of less than 4.

New items are hard to classify as popular or unpopular, due to the sparsity of information about that item. Thus, we examine the user who is the first to rate the new item. We start by extracting the ego network of the user who first rated a new item, to depth three. In other words, we include all first, second and third neighbours of the ego and only allow edges corresponding to ratings made during a certain period of time prior to the first rating of the new item. The aim is to use the least amount of information possible to be able to make future predictions quickly. Analysis of user activity showed that in the case of the MovieLens network, ratings made up to ten days prior to the first rating of the new movie have to be included. Any period less than ten days resulted, in most cases, in an ego network that only contained a single edge. Since the dynamics in the Digg network is much faster, a period of six hours prior to the first rating is sufficient. Since we consider the ego network of the first user, we henceforth refer to this user as the ego. A depth of three is necessary to be able to calculate the local clustering coefficients of the ego.

Since the popularity score ρ of an item is dependent on both the number of ratings as well as the average rating, the two parameters are separately predicted. The popularity scores thus obtained are compared to the actual popularity scores calculated from the real data to assess our predictions.

5.3.2 Predicting the number of ratings

As will be demonstrated below, both the ego's degree, ie. the ego's rating activity, as well as the number of its second neighbours perform poorly as predictors, whereas, the ego's clustering behaviour is a better predictor of the number of ratings that the new item will receive.

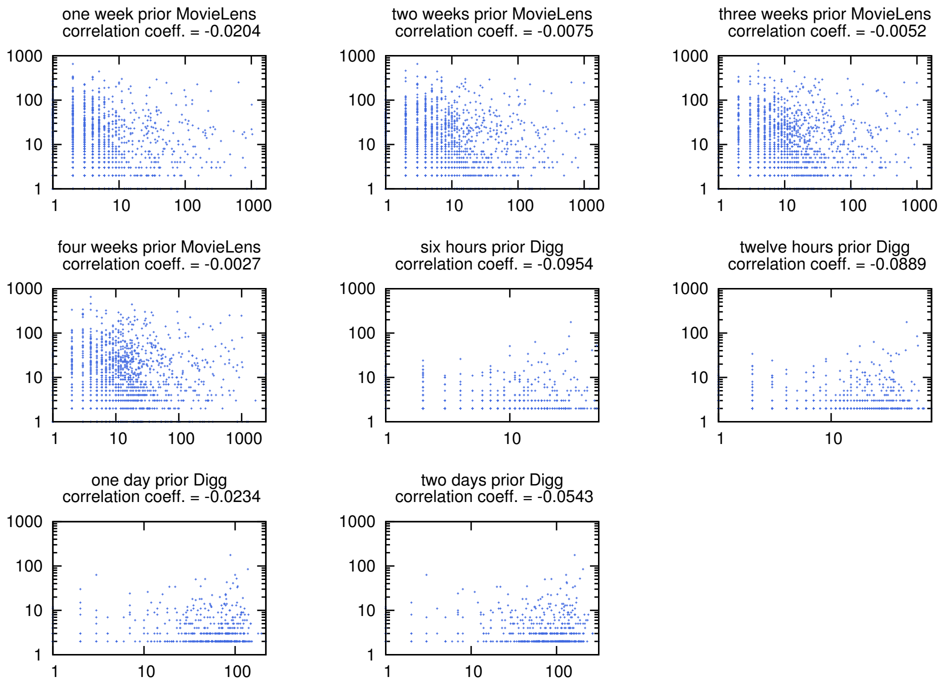


FIGURE 5.9: In order to demonstrate that the ego's degree is not a good indicator for the number of ratings the new item will receive, we plotted the new items' degrees (y-axis) at the end of the critical period against the corresponding ego's degree (x-axis). The first four plots correspond to the MovieLens network, the last four plots correspond to the Digg network. We considered ego degrees one, two, three and four weeks prior to the first rating in the MovieLens. For the Digg network we considered ego degrees six hours, twelve hours, one day and two days prior to the first rating. In all cases the correlation coefficient is approximately zero (see title of each plot).

5.3.2.1 The ego's rating activity

As the previous section showed, a node with high degree is not necessarily influential. This is the case in both the MovieLens and Digg networks. In addition, the rating of a new item by a highly active user does not imply that the item will receive many ratings. Figure 5.9 demonstrates this.

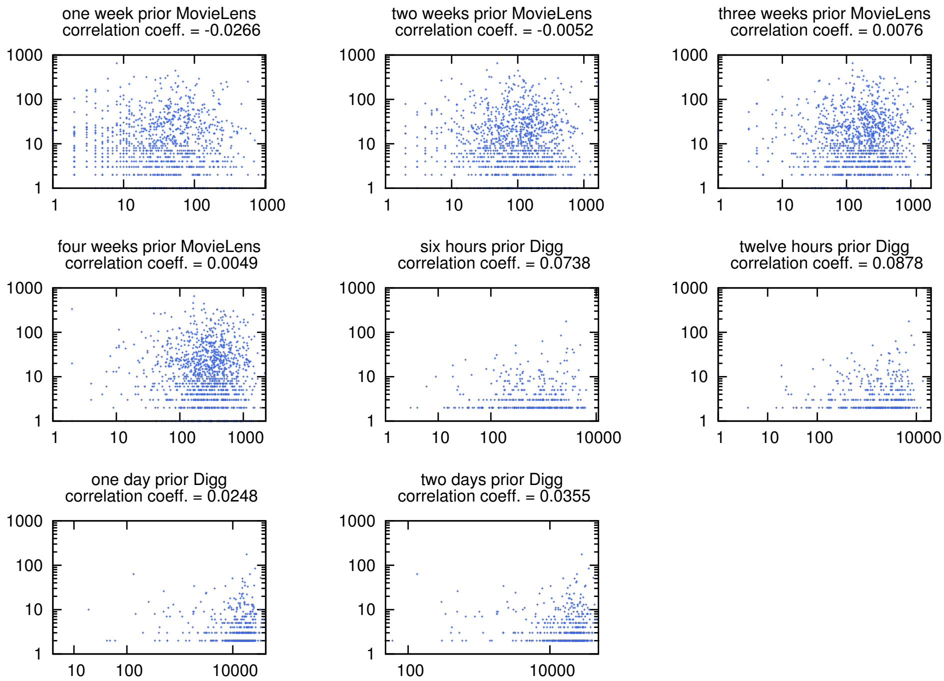


FIGURE 5.10: In order to demonstrate that the number of second neighbours of the ego is not a good indicator for the number of ratings the new item will receive, we plotted the new items' degrees (y-axis) at the end of the critical period against the corresponding number of second neighbours of the ego (x-axis). The first four plots correspond to the MovieLens network, the last four plots correspond to the Digg network. We considered ego degrees one, two, three and four weeks prior to the first rating in the case of MovieLens. For the Digg network we considered ego degrees six hours, twelve hours, one day and two days prior to the first rating. In all cases the correlation coefficient is approximately zero (see title of each plot).

5.3.2.2 Second neighbours of the ego

Since rating networks are bipartite, it is more apt to consider the number of second neighbours of an ego as a predictor of an item's number of ratings instead of the ego's degree, as the latter only gives the the number of items rated by the ego. The second neighbours of the ego are users who rated at least one item that was also rated by the ego. In addition, users who rated the same items as the ego in the recent past are more likely to rate the new item immediately after this user, than a randomly selected user. Hence, the number of second neighbours of the ego, may give some indication of the number of ratings that the new item will receive in the near future.

However, as depicted in Figure 5.10, the number of second neighbours is also a poor indicator for the item's future degree.

5.3.2.3 The ego's clustering behaviour

We now examine the clustering behaviour of users who were the first to rate a new item. Table 5.7 shows that the extracted ego networks vary considerably.

		MovieLens	Digg
size	range	[3, 4411]	[25, 11769]
	mean	1681	1521
	sd	1060	1504
mean degree	range	[2, 540]	[2, 38]
	mean	82	8
	sd	69	4
density	range	[0.0098, 1]	[0.0038, 0.5217]
	mean	0.1065	0.0397
	sd	0.1793	0.0477

TABLE 5.7: The table shows the range, mean and standard deviation of the size, average degree and density of the extracted ego networks.

To determine the ego's clustering behaviour, we calculate the four different local clustering coefficients that we introduced in Chapter 4. As the MovieLens and Digg networks are time independent, we use Equations (4.24) - (4.27).

The calculated clustering coefficients are displayed in Appendix B (see Tables B.3 and B.4).

The four different clustering coefficients measure the proportions of induced 6-cycles, 6-cycles with one chord, 6-cycles with two chords and 6-cycles with three chords respectively. Ugander et al. [123] have shown that nodes with a low local clustering coefficient attract more connections. Hence, we expect that if an ego's clustering coefficient is lower than the average clustering coefficient in its ego network, then the new item will receive a high number of ratings in the near future, as it is likely that new connections are formed. This is very different to the analysis carried out in the previous section, as we are now looking at dynamic networks.

We compare the ego's clustering behaviour to that of all other users in its ego network by calculating how many standard deviations it lies away from the average local clustering coefficient over all users in the ego network. We chose this method over the driving score technique, as the ego networks are relatively large and computing the driving score quickly becomes computationally infeasible. We expect that a high difference in standard deviations indicates that the new item will receive many ratings in the future, provided the ego's clustering coefficient is lower than the average. If on the other hand, the ego's clustering coefficient is higher than the average and the difference in standard deviations is high, we expect the new item to receive very few ratings in the future.

Since the three chord clustering coefficient, icc_3 shows higher connectivity than the induced clustering coefficient icc_0, we give the differences in standard deviations appropriate weights, according to their level of connectivity. The first rating of the new item is also taken into account, since it is likely to influence other users. For instance, if the first rating of a new item is low, it is less likely to receive many ratings than if the first rating is high.

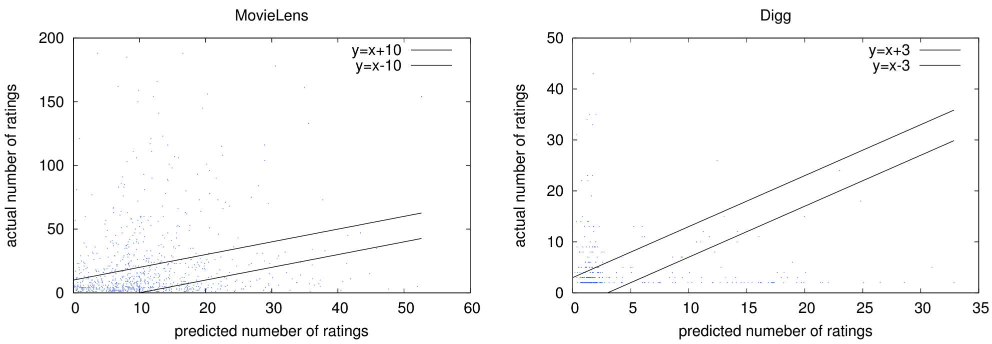
The following equation gives the predicted number of ratings $\hat{n}$ if all four ego's local clustering coefficients are lower than the average:

$$\hat{n} = \frac{r}{3}(2\Delta icc_{ego,0} + 3\Delta icc_{ego,1} + 4\Delta icc_{ego,2} + 5\Delta icc_{ego,3}), \tag{5.6}$$

where r is the first rating of the new item that was given by the ego and $\Delta icc_{ego,k}$ is the difference in standard deviations between that particular clustering coefficient of the ego and the average of the same clustering coefficient in the ego's network. Since ratings range between 1 and 5, the initial rating r is divided by 3. Hence, a rating of 3 is treated as neutral.

If, on the other hand, one or more of the ego's clustering coefficients are higher than the average, then we divide by the corresponding weight instead of multiplying. For example, if in a given ego network, $icc_{ego,0}$ and $icc_{ego,1}$ are lower that the corresponding average clustering coefficient and the other two local clustering coefficients $icc_{ego,2}$ and $icc_{ego,3}$ are higher, then Equation (5.6) becomes: $\hat{n} = \frac{r}{3}(2\Delta icc_{ego,0} + 3\Delta icc_{ego,1} + \Delta icc_{ego,2}/4 + \Delta icc_{ego,3}/5)$.

The reason for only using the first reviewer of an item to predict its popularity is that our aim is to make predictions as early as possible. Considering a combination of the first few ratings may improve predictions, however, it comes with the disadvantage of having to make the predictions later in time.



(A) (B)

FIGURE 5.11: We plotted the actual number of received ratings, n, as a function of the predicted number of ratings, $\hat{n}$. For the MovieLens network (A), we correctly predicted the number of ratings for 53% of the movies. These movies are represented by dots within the two straight lines. For the Digg network (B), we correctly predicted the number of ratings for 57% of news stories.

We have predicted the popularity of all movies that were released between 2004 and 2007 in the MovieLens network. Our method correctly predicted the number of ratings of 510 of the 962 movies (see Figure 5.11A). This is a success rate of approximately 53%. We correctly predicted the number of ratings of 57% of the stories in the Digg network (see Figure 5.11B).

Amongst the movies where the actual number of ratings was lower than predicted, approximately 30% were in languages other than English. These movies generally received very positive reviews by film critics. Hence our predictions overall agree with film critics and the low actual number of ratings in the MovieLens dataset may be explained by the high number of English speaking users. Table B.5 in Appendix B lists some of these movies together with the number of ratings that were collected by the websites Rotten Tomatoes (`http://www.rottentomatoes.com/`) and Metacritic (`http://www.metacritic.com/`). In addition, for the website Rotten Tomatoes the table displays the tomatometer score that represents the percentage of approved critics that have given the movie a positive review. For Metacritic we also show the metascore. The metascore ranges between 0 and 100, with 100 being the best possible score.

Among the movies that our method predicted would receive a lower number of ratings than in reality are many that received very mixed or negative reviews from other websites. Since we do not have any information about the MovieLens users, we are unable to explain these results. It may be possible that in these cases, the user who first rated the movie usually does not watch movies in that particular genre. Another reason may be that these movies were highly anticipated and therefore received many ratings, although scores were generally low.

5.3.3 Predicting the average rating

To estimate the average rating of items, we again make use of the logistic curve. Since ratings lie in the interval $[1, 5]$, the function values should have the same range:

$$\hat{r}(n) = 1 + \frac{4}{1 + e^{-k(n-c)}}, \tag{5.7}$$

where c and k are constants. The constants are chosen such that $\hat{r}(n) \approx 1$ if an item has a predicted number of ratings equal to zero and $\hat{r}(n) = 4$ if an item has a predicted number of ratings equal to the number of ratings that the average item received.

In the MovieLens network the average item received 29 ratings and hence,

$$\begin{aligned}
1.0005 &= 1 + 4/(1 + e^{-k(0-c)}) \\
0.5 \cdot 10^{-3} &= 4/(1 + e^{-k(0-c)}) \\
0.5 \cdot 10^{-3}(1 + e^{ck}) &= 4 \\
e^{ck} &= 7999 \\
ck &= 8.987 \\
c &= 8.987/k
\end{aligned} \tag{5.8}$$

and

$$\begin{aligned}
3 &= 4/(1+e^{-k(29-c)}) \\
3(1+e^{-k(29-c)}) &= 4 \\
e^{-k(29-c)} &= 1/3 \\
-29k+ck &= -1.099 \\
29k &= 10.086 \\
k &= 0.348. \qquad (5.9)
\end{aligned}$$

Therefore, $\hat{r}(n) = 1 + 4/(1+e^{-0.348(n-25.825)})$ for the MovieLens network. In the case of the Digg network, the average rating does not need to be predicted, since we associated every edge with a rating of 5.

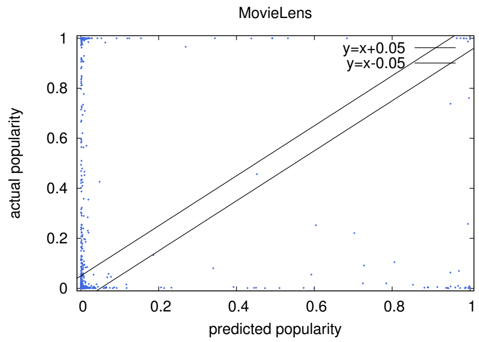
Using Equation (5.7) together with the first rating r and the predicted number of ratings $\hat{n}$, we can estimate the future average rating, $\hat{\mu}$, of the new movie:

$$\hat{\mu} = (\hat{r}(\hat{n}) + r)/2. \qquad (5.10)$$

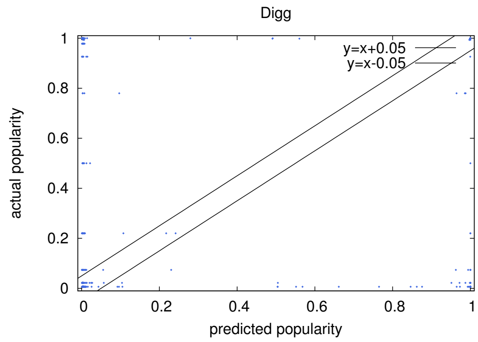
While Equation (5.7) predicts the future average rating well, the rating that is given by the first user has a big influence on other users. Therefore, we take the average between $\hat{r}(\hat{n})$ and the first rating (Equation (5.10)) to improve prediction of the future average rating of an item.

With the two parameters, $\hat{\mu}$ and $\hat{n}$, the popularity score can now be predicted for each item.

We predicted the popularity of all movies that were released between 2004 and 2007 in the MovieLens network and 350 randomly chosen news stories in the Digg network. Figure 5.12 compares the predicted popularity scores $\hat{\rho}$ (Equation (5.3)) to the actual popularity scores ρ and shows that our method correctly predicted the popularity of 638 of the 962 movies in the MovieLens network, with the difference between the predicted popularity $\hat{\rho}$ and the actual popularity ρ being less than 0.05. This is a success rate of approximately 66%. This is especially good since no use was made of any information



(A)



(B)

FIGURE 5.12: We plotted the actual popularity of items in the MovieLens (A) and Digg (B) networks as a function of the predicted popularity. For the MovieLens network we were able to correctly predict the future popularity of 66% of the movies. For the Digg network we achieved a success rate of 51%.

about the new movie other than the first rating. Previously, researchers were only able to predict the popularity of a new movie with 30% accuracy [133]. In the Digg network we achieved a success rate of approximately 51%, where we were able to correctly predict the popularity of 179 out of 350 news stories, compared to the 20% success rate achieved in [133].

5.3.4 Discussion

We showed that the bipartite clustering coefficient may be used as a tool to predict the future popularity of items in rating networks. We focused on improving predictions of new items that are hard to classify as popular or unpopular due to lack of information. Since a new item, at the time of prediction, has degree one, we examined the clustering coefficient of the user who rated the item first. If this user has a low clustering coefficient compared to the users in its neighbourhood it is likely that connections to the new item are formed in the future.

5.4 Summary

In this chapter, we looked at two very different applications of the bipartite clustering coefficient that we introduced in Chapter 4. We utilised the different coefficients to identify important and influential nodes and to predict the popularity of new items in rating networks.

We introduced a novel measure that assigns a score to each node in the network, reflecting whether it is contributing to the global clustering behaviour of the network. Ranking the nodes according to this measure that we call the driving score, allowed us to successfully identify the most influential nodes in two real world networks.

In the second part of this chapter we considerably improved current prediction rates of the popularity of new items in rating networks, using the bipartite clustering coefficients. By investigating the clustering behaviour of the user who was the first to rate a new item we correctly predicted the future popularity of over 65% of new movies in the MovieLens dataset and over 50% of new stories in the Digg dataset. This is a major improvement over previous research.

Note that predicting the popularity of an item that has not been rated (ie. an isolated node) is impossible when the only source of information is the network's topology. Once the new item is connected to one user, we can look at the user's neighbourhood and its clustering behaviour and thus make predictions about the future popularity of the new item. The reason for looking at only the

first user who rates the item, as done in this chapter, is that websites wish to be able to make predictions as soon as possible, that is in our case when the new item has been rated the

first time. The question arises of whether the results would be improved by considering more than one user.

Chapter 6

Crime Networks

Part of this chapter has been published in [68].

6.1 Introduction

This chapter brings together the measures and techniques introduced in the previous chapters by presenting the case studies of two crime networks.

6.1.1 Motivation

The case studies presented here are motivated both by the difficulties in understanding the dynamics of criminal activity and recent research showing that presenting crime data in the form of complex networks leads to useful insights into the dynamics of crime [30]. The analysis of crime data is crucial for prevention and assessment of illegal activity.

Crime does not occur uniformly across different locations and time. For example, a property that has been burgled once is at higher risk of being burgled again in the near future as are properties in its immediate neighbourhood [52]. Such locations that experience higher crime rates are commonly called hotspots [117]. Identification of current hotspots and prediction of future hotspots would allow a more effective allocation of police resources [26]. The existence of hotspots shows the importance of considering the spatio-temporal nature of crime networks when searching for significant patterns in criminal activity.

6.1.2 Outline

Our contributions in this chapter are the following: We present two case studies of bipartite, spatio-temporal crime networks to demonstrate the potential of the measures and techniques (that we introduced in the previous chapters). This chapter describes ongoing work raising several questions directly relevant to the analysis of crime networks that need to be answered in future work, to be able to advance the understanding of the dynamics of criminal activity. Analysing crime networks in their original bipartite structure by applying our novel techniques that we introduced in the previous chapters, leads to new insights that have been missing in previous research and raises questions that need to be answered in future research.

The chapter is structured as follows: The first case study is presented in Section 6.2. It examines crime data collected in the state of New South Wales, Australia. Section 6.3 presents a case study of crime data that was collected in the United Kingdom. We conclude this chapter with a summary in Section 6.4.

6.2 Case study I: The New South Wales crime dataset

This section demonstrates the potential of the techniques that we introduced in the previous chapters and highlights possible future directions for research in the area of crime prevention.

The New South Wales crime dataset is publicly available at `http://data.gov.au/dataset/nsw-crime-data/` and contains information about the different types of crime that took place in the state of New South Wales, Australia, between January 1995 and December 2012. It records the local government area where a crime occurred along with its offence category and the month and year of the crime. The New South Wales Bureau of Crime Statistics also provides a helpful visualisation tool for the dataset on their website (see `http://crimetool.bocsar.nsw.gov.au/bocsar/`). This website allows the user to research various basic statistics of the local government areas and offence categories. More details can be found in Chapter 2 (see Subsection 2.4.5).

The New South Wales (NSW) crime data contains 155 local government areas and 62 offence categories that we represent by primary and secondary nodes respectively. This bipartite network is time independent as a crime can occur at any point in time. We are particularly interested in changes in the data over time and hence have divided the dataset into 216 networks, each covering a period of one month. Analysing each network separately and comparing the results gives valuable insights into the dynamics of criminal activity with respect to the local government areas.

6.2.1 Co-occurrence of crimes

Crimes often co-occur [92]. For instance, a person who breaks into a house may be charged with theft, trespass and assault. To find crimes that co-occur at a significantly higher rate, we project the bipartite networks constructed above onto the set of offence categories and then extract the backbones (see Definition 3.2). An offence category is connected to a local government area if the type of crime occurred at least once in that area in the given time period. Extracting the backbone of the projection onto the offence categories for each of the 216 months allows the identification of crimes that co-occur at a significant rate and their change over time. Identification of co-occurring crimes would allow the implementation of better prevention techniques as crimes may be now linked to each other. For example, reducing the rate of burglaries may simultaneously reduce other crimes that co-occur.

Finding crimes that co-occur in this manner has many advantages over performing a simple correlation. For example, it allows to easily compare the significance of two pairs of crimes, since backbone extraction considers the network as a whole. This is not possible for two correlation coefficients obtained by comparing two different pairs of crimes. Further we noticed that some correlations, which the literature reported as strongly correlated, were rather weak.

We extracted the backbone of the projection onto the set of offence categories for each month between the years 1995 and 2012 and compared the change in significance of connections between all 1,891 pairs of crimes. Below we look at a few of the 62 offence categories, including categories related to property crime and domestic violence.

Note that we consider an edge significant according to Definition 3.3. Hence, edges are included in the backbone if their weight is greater than the mean of the approximated weight distribution plus three standard deviations.

6.2.1.1 Property crimes

Property crimes are one of the most common crimes in NSW [3] and include the offence categories: *break and enter dwelling/non-dwelling*, *malicious damage to property*, *motor vehicle theft*, *steal from dwelling*, *trespass*, *arson*, etc.

Crimes falling in the category *break and enter dwelling* form highly significant connections with the categories *domestic violence related assault*, *harassment, threatening behaviour and private nuisance*, *non-domestic violence related assault* and *break and enter non-dwelling*, with significance levels being above 100 standard deviation during each month between January 1995 and December 2012 (see Figure 6.1).

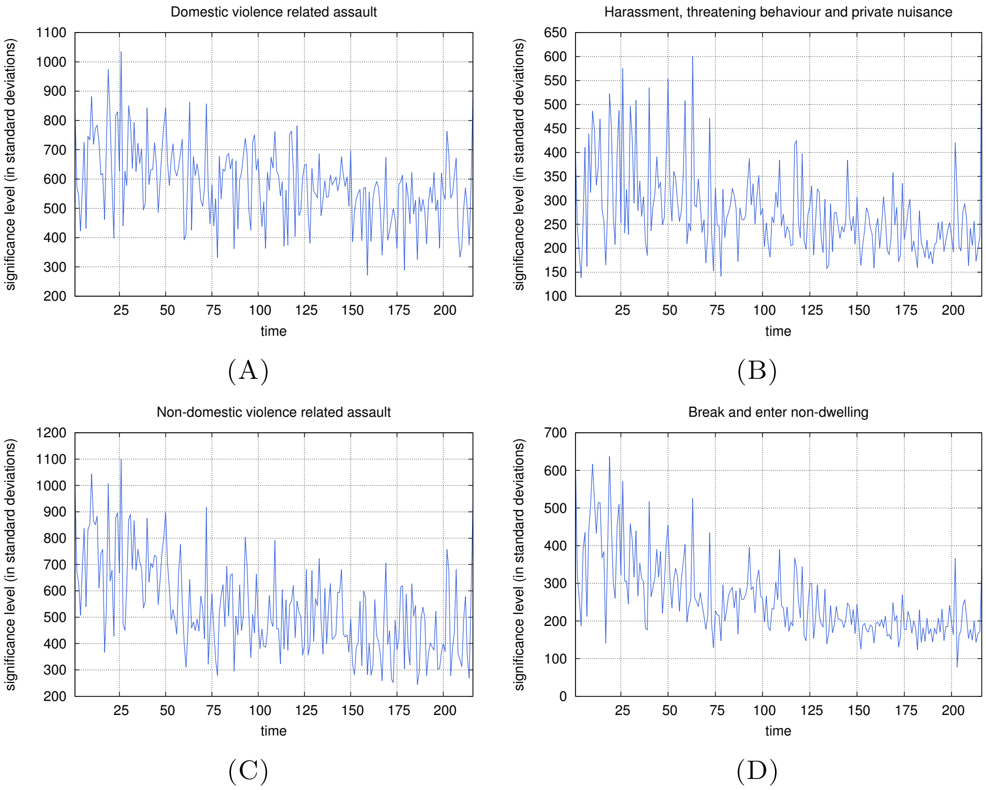


Figure 6.1: Significance levels over time for the connections between the category *break and enter dwelling* and the categories (A) *domestic violence related assault*, (B) *harassment, threatening behaviour and private nuisance*, (C) *non-domestic violence related assault*, and (D) *break and enter non-dwelling*.

Significant connections are also formed between the offence category *break and enter dwelling* and most other offence categories. The same holds for the offence category *break and enter non-dwelling*. This observation indicates that crimes falling into the two categories *break and enter dwelling* and *break and enter non-dwelling* are closely related and hence local government areas experiencing high rates of residential break-ins also experience high rates of non-residential break-ins.

Interestingly, the offence category *malicious damage to property* shows the most significant connections to *drug and liquor related offences* and more expectedly to the categories *trespass* and *arson*. The first observation is confirmed in [50], stating that offenders who are charged with malicious damage to property are often intoxicated.

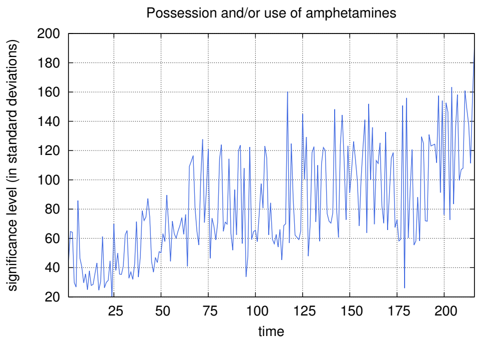
6.2.1.2 Domestic violence related crimes

According to the Australian Bureau of Statistics, domestic violence impacts a large proportion of the NSW population. In this subsection, we identify the most significant connections between domestic violence related crimes and other crimes over time.

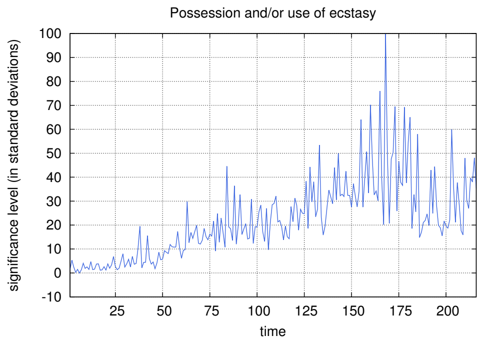
We find that domestic violence related assault forms significant connections to most other offence categories (approximately 75% of all categories) between the years 1995 and 2012. This observation shows that domestic violence is ubiquitous throughout NSW and generally co-occurs with additional crimes. Studies have shown that substance abuse and family violence often co-occur [31]. Indeed, we observe an upward trend in the significance of connections between domestic violence related assaults and drug related crimes (see Figure 6.2).

Interestingly, we can also see an upward trend in the significance of connections between domestic violence and pornography offences. Between 1995 and 2000 the significance levels fluctuated between zero and five standard deviations, whereas significance levels in 2012 are as high as approximately 35 standard deviations. In fact, NSW police believe that there is a link between the two offences with NSW Police assistant commissioner Mark Murdoch saying in December 2014 that pornography is one reason for an increase in domestic violence offences [97].

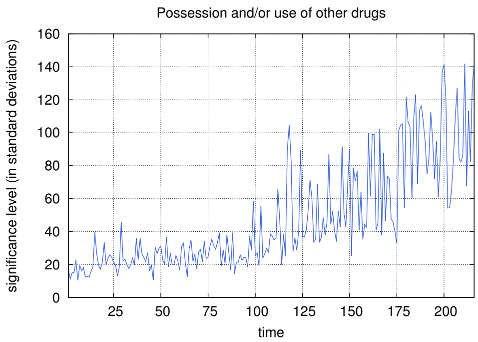
The offence category *domestic violence related assault* also shows highly significant connections with the offence categories *breach apprehended violence order*, *breach bail conditions*, *sexual assault* and *indecent assault*. For the category *breach bail conditions* significance levels were higher than 100 standard deviations throughout the 18 year period. The category *breach bail conditions* shows significance levels of approximately 60



(A)



(B)



(C)

FIGURE 6.2: There is a clear upward trend in the significance of connections between domestic violence related assaults and drug related crimes.

standard deviations in 1995 with a clear upward trend. The two categories *sexual assault* and *indecent assault* show highly fluctuating significance level, the lowest being approximately 75 standard deviations and the highest being approximately 1100 standard deviations in both cases.

The above results lead us to the following question:

Question 6.1. *How can we predict whether different crimes will co-occur at a significant rate in the future and how would this information help in the implementation of crime prevention measures?*

A possible way of answering this question is the thorough examination of the development of the backbone over time. It would be interesting to study whether the absence of particular connections in the backbone affect the significance of other connections.

6.2.2 Areas similar in crime

In this section we apply our backbone extraction technique to divide the local government areas of NSW into communities. This is a first step towards the improvement of crime prevention strategies. Certain strategies of crime prevention that are already in place in some areas may then be applied to other areas, with the caveat that a prevention scheme that works in one location is not guaranteed to be successful in another. If two local government areas have a significant connection as identified by our backbone extraction method, they would experience similar criminal behaviour, leading to the following question:

Question 6.2. *Are prevention strategies that work well in one area more likely to work other areas that are part of the same community?*

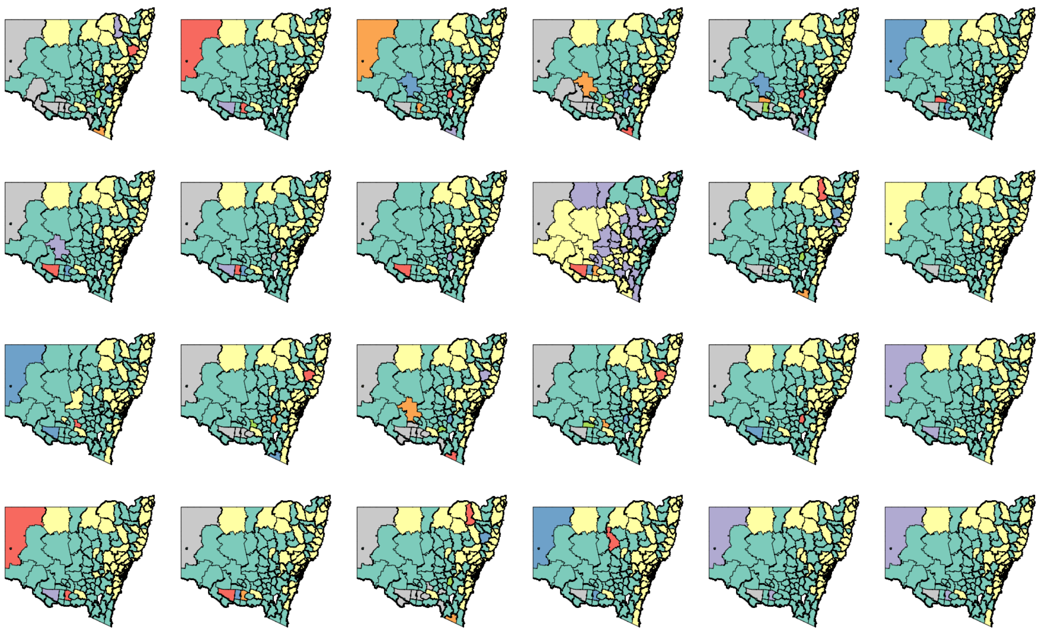


FIGURE 6.3: Maps of New South Wales and its government areas in the months January 1995 - December 1996. The different local government areas are coloured according to their group membership. The colour grey indicates no data being available for the corresponding area in that month.

To identify communities of local government areas, we extract the backbone of the projection onto the different areas for each month between January 1995 and December 2012. In each case we found two large communities as well as some smaller communities. The two largest communities usually contained government areas in the north east and south west respectively. Figure 6.3 shows a map of NSW and its local government areas between January 1995 and December 1996. Figure 6.4 shows a map of NSW and its local government areas between January 2011 and December 2012. The two sets of maps were chosen at random - for illustration purposes. Areas are coloured according to community membership. The largest community is coloured in green, the second largest in yellow, with the size of a group determined by the number of its members and not the total area covered. The colour grey represents missing data in the respective month.

FIGURE 6.4: Maps of New South Wales and its government areas in the months January 2011 - December 2012. The different local government areas are coloured according to their group membership. The colour grey indicates no data being available for the corresponding area in that month.

We could not identify a large degree of variability in the two largest communities over time. This suggests that the dynamics in criminal activity responsible for the formation of clustered government areas are stable over time. During each month we found at least one community that consisted of a single government area. In addition, some government areas tended to change communities. This observation brings us to the following questions:

Question 6.3. *What is the cause of some of the local government areas frequently changing communities?*

Question 6.4. *Why is the community structure of the projection onto the set of local government areas stable over time?*

To answer the above questions, we need to examine the individual government areas more closely. This task is left for future research.

6.2.3 Clustering behaviour of crime networks

In this subsection we demonstrate that the bipartite clustering coefficients of crime networks is related to the severity of the different offence categories and the density of population in the different local government areas of NSW.

We calculated the four local time independent clustering coefficients (see Equations (4.28) - (4.31)) with respect to each of the local government areas and offence categories and compared them to the average local clustering coefficients in the complete network. In order to compare the individual local clustering coefficients to the network's average, we calculate the Z-score and normalise to unity to reflect the difference in standard deviation of the various coefficients to the mean of the network. A score close to zero shows that the local clustering coefficients of a particular node are higher than the mean local clustering coefficients of the network. A score close to 0.5 shows a similar clustering behaviour to the average, and a high score (close to one) represents a clustering behaviour that is much lower than the average.

6.2.3.1 Ranking offence categories

After calculating the normalised Z-scores, we are able to rank the 62 offence categories. We observe that more common, often less serious, crimes are ranked low (close to zero) while, offences that are less common, but more serious, are given a high rank (close to one).

The ranking of offence categories changes from month to month, however, the observed difference in the ranking of each category is generally small. Two of the lowest ranked categories in the NSW crime dataset between the years 1995 and 2012 are *possession and use of cannabis* and *sexual offences*. Some offences that fall under *disorderly conduct* and certain *offences against justice procedures* were also ranked low throughout the 18 year period. Figure 6.5 shows the change in ranking of these offences together with other similar offences that fall within the same super-category.

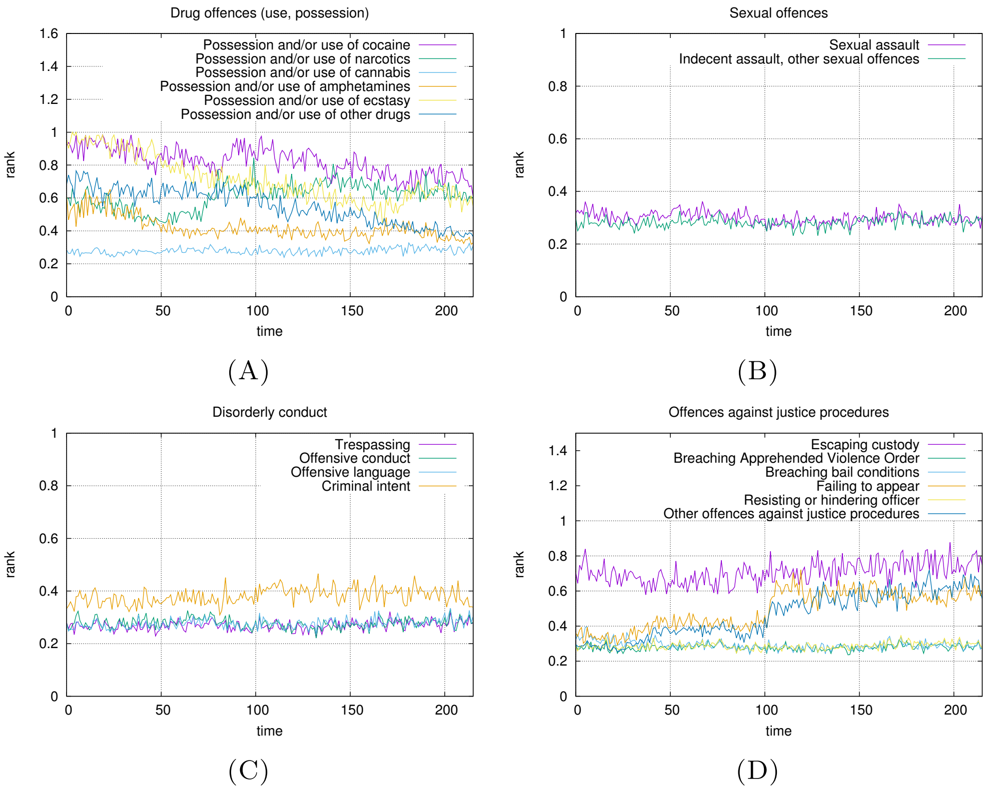
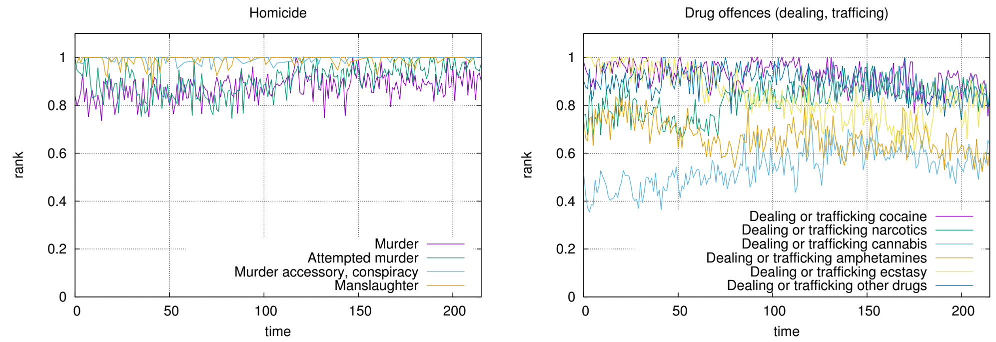


FIGURE 6.5: The rankings of offence categories that were particularly low over time together with similar categories that fall into the same super-category.

Looking at the first plot in Figure 6.5, we can see that the rank of the offence category *use or possession of cannabis* is much lower than that of other illicit drugs. Although Australia has seen a significant decline in the use of drugs after the tightening of drug strategies in 1998, cannabis is still one of the most common and frequently used drugs [124].

Both sexual offence categories recorded in the dataset received very low rankings throughout the 18 year period. Sexual offences are a huge problem everywhere in Australia with New South Wales having the highest total number of sexual assaults reported to police [5]. According to the Australian Bureau of Statistics (`http://www.abs.gov.au/`), 20% of women and 5% of men over the age of 15, experience sexual violence.

Disorderly conduct is another common offence in NSW, specifically on weekends and in connection with alcohol consumption [121]. Interestingly, the category *criminal intent*, is ranked higher than other acts of disorderly conduct. This is an indicator that in many cases the police do not pick up the planning of criminal activity.



(A) (B)

FIGURE 6.6: The highest ranked offence categories were homicide and drug dealing offences.

On the other hand, *homicide* and the *dealing of cocaine* are two of the highest ranked categories (see Figure 6.6). According to the Australian Institute of Criminology [4], homicide incidents are currently one of the lowest crime rates in Australia and it is unlikely that a homicide remains unreported, as is often the case with domestic violence. With regards to cocaine dealing, between 2003 and 2012 cocaine arrests have accounted for less than 1.5% of national illicit drug arrests [2].

Clearly, the rank of offences reflects the severity of the crime. All data indicates that more petty crimes such as *trespassing* occur more often than serious crimes such as murder. We hence arrive at the following questions:

Question 6.5. *Does the ranking of offence categories in countries with a high rate of severe crimes such as homicide, reflect the severity of crimes?*

Question 6.6. *Can we predict the future rate of a particular crime?*

Question 6.5 can be answered by obtaining and analysing crime data from countries with a high rate of for example homicide. To answer Question 6.6, we need to investigate whether the ranking of offence categories or possibly the backbone contains information about the rate at which crimes occur.

6.2.3.2 Ranking local government areas

A total of 155 local government areas form the Australian state of New South Wales. Examination of our results shows that the rank of any individual area never fell below 0.3, meaning that the concentrations of local 6-cycles are skewed with many areas exhibiting concentrations below the average. We found that isolated and sparsely populated areas received extremely high ranks, showing close to no variation over time. We have plotted the ranks of four government areas over time in Figure 6.7.

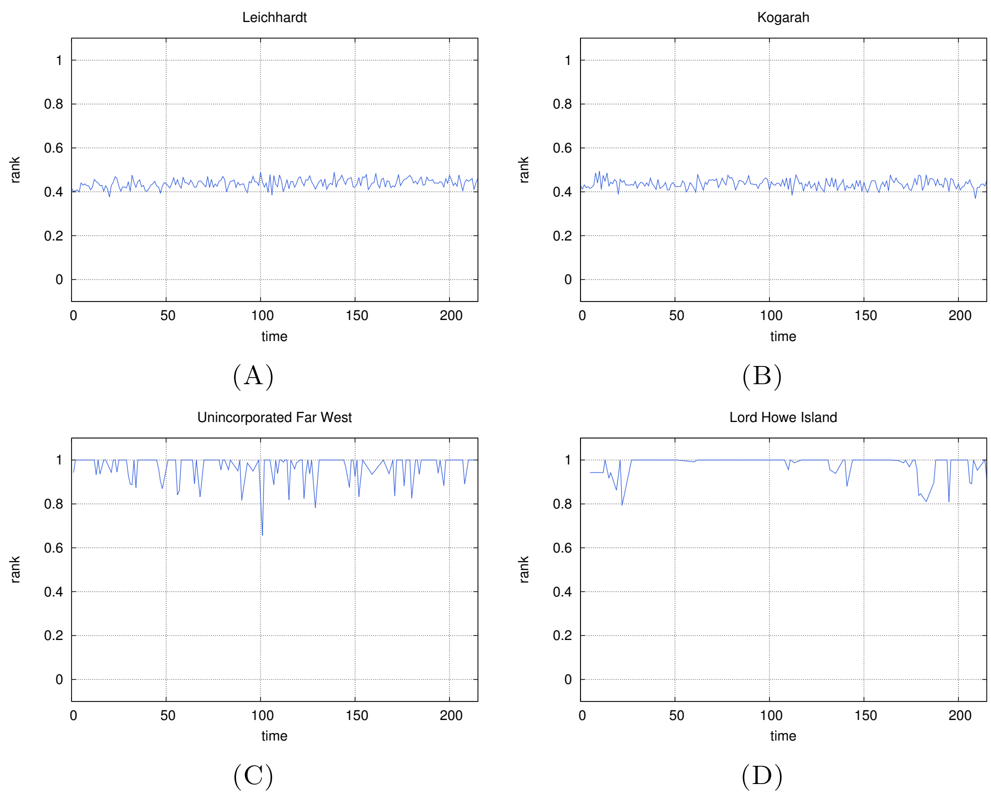


FIGURE 6.7: The graphs illustrate the rankings of four local government areas (Leichhardt, Kogarah, Unincorporated Far West, Lord Howe Island) over time.

We would like to answer the following question in future work.

Question 6.7. *How do the ranks of offence categories and the ranks of local government areas relate to each other?*

6.2.4 Discussion

The analysis of the NSW crime network, using backbone extraction and the bipartite clustering coefficients, revealed interesting information, much of which we confirmed by researching news articles. Although some of the uncovered information is known by means of other research, it is fascinating that the backbone extraction and application of the bipartite clustering coefficients pick this up by considering only the network's topology.

The analysis of the NSW crime data identified many gaps in the literature and has led to several questions that need to be addressed in future work.

6.3 Case study II: The United Kingdom crime dataset

Crime data from the United Kingdom is publicly available at `https://data.police.uk/` and updated on a monthly basis. It records the approximate location of crimes in form of latitude and longitude coordinates. In addition, the type of crime is recorded, however, the categories are very broad. Each crime is associated with a time stamp that indicates the month in which the crime occurred. Here, we use burglary data collected between January 2016 and June 2016 inclusively.

The U.K. crime dataset is in many ways different to the data that was collected in New South Wales. The U.K. dataset records the longitude and latitude of the crimes, as opposed to the NSW crime dataset that only records the local government areas. On the other hand, the offence categories are more detailed in the NSW dataset. The U.K. data summarises many smaller crime categories. For example, all violence related crimes and sexual offences fall within the same category.

Here we critically review a study presented by Davies and Marchione [26] in which several motifs, sub-graphs that occur with a higher probability in an observed network than in a similar random network [77], are identified in a burglary and a piracy network. In Chapter 3 we demonstrated that cliques (see Definition 2.7) are overrepresented in one-mode projections, leading to our hypothesis that some of the motifs identified in [26] arise from the particular way in which the crime networks were created.

Davies and Marchione [26] created so-called event networks, where a crime (the event) is connected to another crime if they are close in space and time. Two burglaries for instance, may be considered close if they took place within a distance of at most 500 metres of each other and within a time period of four weeks. This way of creating the event networks is similar to the process of one-mode projection. Consider a secondary node in a bipartite network that is associated with a particular area of some city, and a particular time window. Crimes, represented by primary nodes, are connected to a particular secondary node if they occurred in the secondary node's associated area and time window. Hence, when projecting the bipartite network, all crimes connected to a particular secondary node would be connected to each other, forming a clique. This is also the case for the creation of event networks, leading us to question whether the sub-graphs identified in [26] are motifs. In fact, most of the sub-graphs identified as occurring at a significantly higher rate are cliques. This may possibly be the result of the network creation that is essentially a one-mode projection.

The identification of motifs requires comparison to an ensemble of random networks. Generally, the configuration model (see Subsection 2.2.5) is a suitable random graph model for the ensemble. In the case of spatio-temporal networks however, the configuration model is inappropriate as events that are distant in space and time may be connected when the network is rewired. In essence Davies and Marchione [26] propose a technique to generate random spatio-temporal networks which may be subject to the problems identified in previous chapters.

The authors of [26] generate the random networks by creating events that are placed uniformly at random in space and time. The events are then shifted in space and time to ensure that the random networks have the same number of edges. In other words, the final random network has the same number of pairs of nodes that are close in space and time. The question of whether this process of modifying the event networks guarantees an unbiased sample remains unanswered and needs to be addressed. In addition, randomising the one-mode networks is likely to bias the occurrence of particular sub-graphs (see Subsection 4.3).

We chose the U.K. crime dataset to test whether our hypothesis of the occurrences of sub-graphs being biased in event networks holds true for several reasons. Most importantly, the U.K. crime dataset is in many ways similar to that studied in [26]. Since we wish to compare our findings to the study presented in [26] and because we were unable to obtain the same dataset, the U.K. dataset is our first choice. Secondly, the U.K. crime dataset records the longitude and latitude coordinates of the different crimes, necessary for the construction of event networks similar to those in [26].

The NSW dataset on the other hand, records crime locations at a much coarser scale, making the data unsuitable for comparison. In addition, the dynamics of criminal activity in Australia may be very different to the dynamics of criminal activity in the United Kingdom.

6.3.1 Motifs in spatio-temporal networks

The burglary event network as created in [26] is generated in much the same way as a binary one-mode projection. Due to the fact that projections are dense networks and lead to the over-representation of cliques, we believe that the notion of motifs may be biased in event networks. In other words the high concentration of a particular sub-graph may arise from the manner in which the event network is created.

To test our hypothesis, we create several burglary networks where each primary node corresponds to a different burglary and each secondary node is associated with a geographic area and a time period. A primary node is connected to a secondary node if the crime was committed in the area and time period that corresponds to the secondary node. We then project the created network onto the set of crimes and count the number of sub-graphs of size three and four. Next, we compare the observed count to the expected count in similar random networks.

In order to generate an unbiased ensemble of random networks, sampled uniformly at random, we randomise the initial bipartite network using the Curveball algorithm (see Subsection 2.2.5.2) and then project it onto the set of crimes. Randomising the bipartite network ensures that the identification of motifs is not biased by the process of projection. Further, the spatial and temporal constraints discussed in [26] are also met.

For the enumeration of the different sub-graphs in the observed network and the ensemble, we employ the QuateXelero algorithm [54]. The algorithm is fully described in Chapter 7 (see Subsection 7.2.2). Note that we use the QuateXelero algorithm to count the sub-graphs, but not to randomise the observed network. We randomise the networks as described above.

6.3.1.1 The observed network

The data we analyse in this subsection covers the area of Greater London and records burglaries in that area between January and June 2016 (see Figure 6.8).

Since the crimes do not have exact time stamps, but are associated to the months in which they occurred, we split the data into six smaller sets each covering a period of one month. For each month we generate the secondary nodes by creating evenly spaced geographical

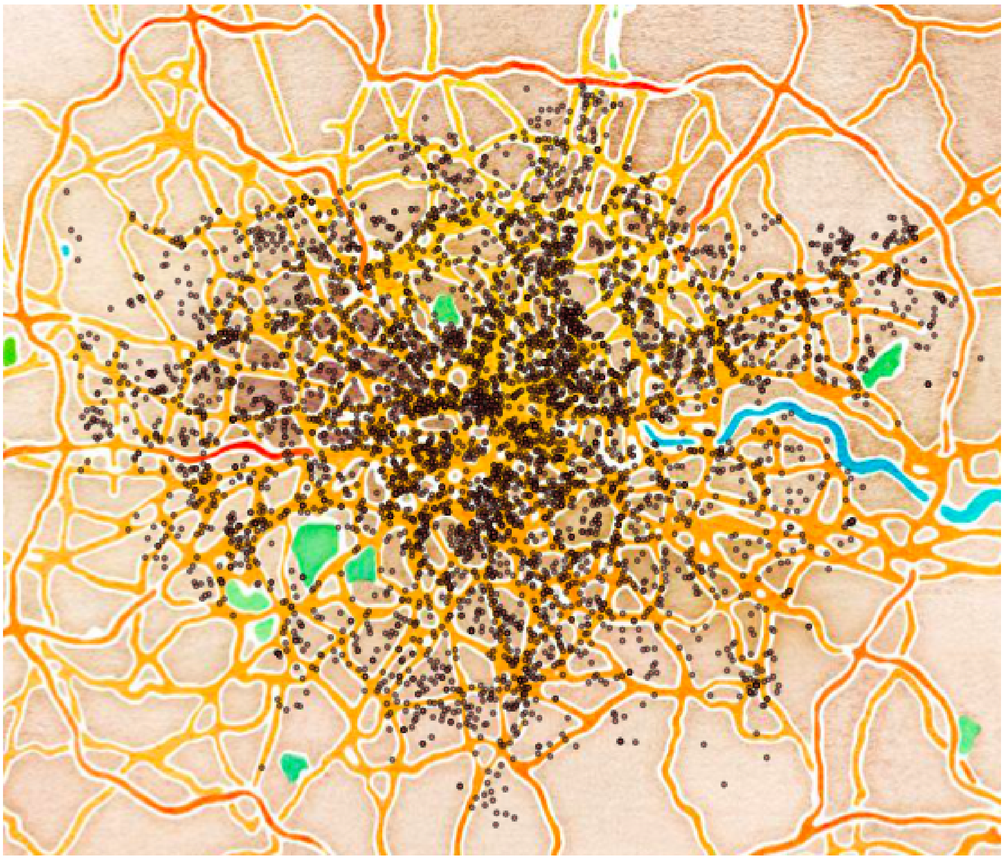


FIGURE 6.8: Incidences of burglaries in Greater London in June 2016

points that cover the area of Greater London. A particular instance of burglary is connected to a secondary node if it occurred within 300 metres of the secondary node's location. Hence, when projecting onto the set of crimes, two crimes that occurred within a distance of more than 600 metres cannot be connected.

Table 6.1 shows the number of different sub-graphs of order three and four that we counted in the projection onto the set of burglaries.

6.3.1.2 Comparison to the ensemble of random network

We now compare the number of the different sub-graphs in the observed networks to the average number in the ensemble of random networks. We created 100 random networks for each month between January and June 2016 inclusive by randomising the original bipartite networks using the Curveball algorithm (see Subsection 2.2.5.2) and enumerated the different sub-graphs using the QuateXelero algorithm [54]. Table 6.2 shows the average number of different sub-graphs of order three and four as counted in the ensemble

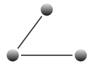
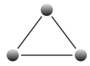
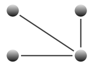
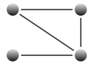
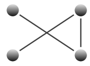
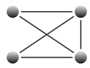
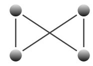
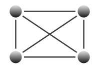
	January	February	March	April	May	June
	11152	8466	8170	5728	4784	7172
	19439	14070	15038	11556	9668	12835
	0	0	0	0	0	0
	84463	57618	65881	30110	20224	42809
	15801	11633	10250	5705	4034	8883
	5682	3777	3928	1804	1755	4147
	22	8	2	0	17	5
	44544	27747	36002	19920	13858	23130

TABLE 6.1: The number of sub-graphs of order three and four as counted in the projections onto the set of burglaries.

of random networks. In addition, the table displays the standard deviation for each sub-graph count.

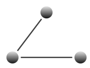
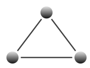
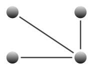
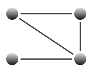
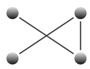
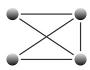
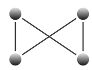
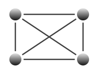
	January	February	March	April	May	June
	11827 (371.6)	9117 (296.1)	8617 (335.6)	6287 (192.1)	5545 (198.1)	7818 (217.1)
	19603 (39.4)	14122 (10.9)	15071 (6.3)	11612 (12.6)	9719 (10.8)	12879 (11.4)
	0 (n/a)	0 (n/a)	0 (n/a)	0 (n/a)	0 (n/a)	0 (n/a)
	86638 (4516.2)	58666 (3713.6)	65246 (4688.2)	32058 (1706.7)	23681 (1354.3)	46525 (2207.5)
	16307 (1522.7)	12051 (1109.7)	10435 (1279.6)	6178 (543.2)	5067 (695.5)	9681 (917.0)
	4164 (433.9)	2664 (248.7)	2710 (462.4)	1252 (234.3)	1284 (126.7)	2951 (385.6)
	13 (6.7)	6 (3.5)	3 (3.6)	1 (2.0)	10 (5.8)	5 (3.5)
	44885 (87.7)	27770 (7.6)	36009 (2.3)	19954 (11.2)	13881 (6.4)	23148 (7.4)

TABLE 6.2: The number of sub-graphs of order three and four as counted in the ensemble of random networks. The corresponding standard deviation is shown in parenthesis.

6.3.2 Discussion

The above results allow us to calculate a Z-score for each of the different sub-graphs. The Z-score gives the difference in standard deviations between the number of sub-graphs in the observed network and the average number of sub-graphs in the ensemble of random networks. The Z-scores are listed in Table 6.3, revealing that over the six months only the sub-graph in row six of the table is overrepresented.

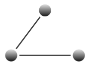
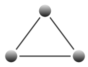
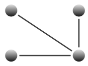
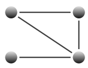
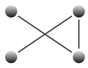
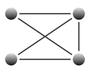
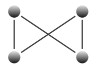
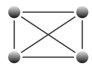
	January	February	March	April	May	June
	-1.8175	-2.1984	-1.3313	-2.9075	-3.8426	-2.9779
	-4.1712	-4.7527	-5.2222	-4.4932	-4.7169	-3.8664
	n/a	n/a	n/a	n/a	n/a	n/a
	-0.4815	-0.2821	0.1354	-1.1416	-2.5523	-1.6833
	-0.3325	-0.377	-0.1445	-0.8716	-1.4846	-0.8701
	3.4973	4.4759	2.6343	2.3545	3.7193	3.101
	1.2791	0.5693	-0.3613	-0.7148	1.1996	0.0854
	-3.8919	-2.9878	-2.8862	-3.0529	-3.5111	-2.3874

TABLE 6.3: The Z-scores of the different sub-graphs.

Clearly, our analysis identified only one of the sub-graphs as a motif, as opposed to Davies and Marchione [26], who identified several. The motif identified by us was also identified in [26], together with the cliques of order three and four. All other sub-graphs that were identified to be motifs in [26] are of order four and contain at least one clique of order three. Although the sub-graph identified by us (see row six of Table 6.3) also contains cliques of size three, it cannot be biased by the projection since we randomised the bipartite network first and then performed the projection. Further it cannot be a result of smaller sub-graphs that are overrepresented since none of the sub-graphs of order three were identified as motifs.

Our analysis confirmed our hypothesis that the identification of motifs in event networks is highly likely to be biased.

6.4 Summary

This chapter presented case studies of two different crime networks. The first case study, looking at crime data from Australia, served as a demonstration of the potential of our techniques for the analysis of bipartite networks, introduced in earlier chapters. The analysis of the NSW dataset raised many questions that need to be answered in future research to be able to advance the understanding of the dynamics of criminal activity.

The second case study looked at burglary networks in the United Kingdom. We compared our results to a previous study that identified several motifs in such networks. We were able to confirm our hypothesis that many of the motifs are a result of one-mode projection.

Chapter 7

Enumeration of Subgraphs

7.1 Introduction

7.1.1 Motivation

The enumeration of sub-graphs in general and paths in particular is a challenging problem of high theoretical interest to mathematicians that finds many applications in a variety of scientific disciplines.

Many network measures, one-mode and bipartite alike, require the enumeration of particular sub-graphs. For example, the bipartite clustering coefficient (see Chapter 4) requires the counting of 6-cycles and paths of lengths four and five. The measure of, for instance, betweenness centrality requires the enumeration of all shortest paths between any two nodes of the network.

Another interesting notion in network science is motifs, sub-graphs that appear with a significantly higher frequency in the observed network than in a similar random network [77]. A vast amount of software for motif detection is available, however, most of them are slow and computation time grows quickly as the network becomes larger.

7.1.2 Outline

In this chapter we describe two of the fastest motif detection algorithms and modify one of these algorithms to enumerate the sub-graphs needed for the calculation of the bipartite clustering coefficients (see Chapter 4). Furthermore, we present preliminary results on the enumeration of paths on the square lattice.

The chapter is structured as follows: Section 7.2 reviews two of the fastest motif detection algorithms, G-tries and QuateXelero. In Section 7.3 we modify the G-tries algorithm to enumerate only the sub-graphs that are required to calculate the bipartite clustering coefficients that we introduced in Chapter 4. Section 7.4 is dedicated to the problem of enumerating paths on the square lattice. The chapter concludes with a summary in Section 7.5.

7.2 Motif detection algorithms

The notion of motifs has received much attention in the network science literature. A motif is a sub-graph that appears in a given network with a significantly higher frequency than in similar random networks [77]. Motifs can be thought of as the building blocks of complex networks. In biological networks small sub-graphs that appear at a significantly high rate often have special functions in the network.

We are not interested in finding motifs in bipartite networks per se, but rather in modifying and improving existing motif detection software to enable faster computation of the bipartite clustering coefficients introduced in Chapter 4.

Enumeration of sub-graphs is a computationally hard problem. The next two subsections describe two of the fastest motif detection software available, one of which we modify to efficiently calculate the bipartite clustering coefficients.

7.2.1 G-tries

Ribeiro and Silva [108] introduce a novel data structure called g-trie, short for **g**raph re**trie**val, based on prefix trees to store and enumerate sub-graphs in a network. All children of a node in a prefix tree have a common prefix.

A g-trie is a multiway tree where, similar to a prefix tree, the children of a node have a common sub-graph. The root of any g-trie is the empty graph. When enumerating the sub-graphs of a simple network, the root of a g-trie would have exactly one child, representing the graph with one node and no edges. The children of any node in a g-trie represent sub-graphs that have exactly one more node than the sub-graph represented by the parent. In addition, each g-trie node contains information about the connectivity of the new node in form of a binary string. Figure 7.1 shows an example of a g-trie. The g-trie node that represents the sub-graph consisting of a single node is represented by the binary string 0 as there is no edge connecting the node to itself. The sub-graph that consists of two nodes is represented by the binary string 10. The 1 in the first position of the string indicates that the node coloured in black is connected to the first node. The 0 in the second position of the string indicates that the node coloured in black does not have a loop.

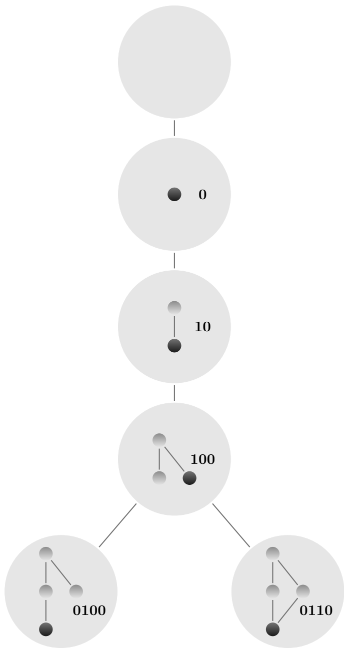


FIGURE 7.1: An example of a g-trie, storing two sub-graphs of order four, represented by the two leaves of the g-trie. The new node that is added to the previous sub-graph is coloured black. The binary strings give information about the connectivity of the new node to existing nodes.

To ensure that isomorphic sub-graphs (see Definition 2.9) are represented by the same leaf of the g-trie Ribeiro and Silva [108] use a canonical labelling of sub-graphs, with the aim to reduce the order of the g-trie. Note that the order of the g-trie directly depends on the choice of canonical labelling. The ideal labelling of sub-graphs is achieved by assigning a low index to a node that has many neighbours. Hence, when the network is searched for a matching induced sub-graph, there are less candidate nodes (assuming the network is sparse) and thus computation time is reduced.

Using the g-trie in the current form to search a network for a set of given sub-graphs can result in the same sub-graphs being matched multiple times. In fact, every sub-graph is found as many times as there are elements in its group of automorphisms.

Definition 7.1. An *automorphism* is a bijection $\varphi : U \to U$ from a graph $\mathcal{G} = (U, E)$ to itself, such that $(u_i, u_j) \in E \Leftrightarrow (\varphi(u_i), \varphi(u_j)) \in E$, $\forall u_i, u_j \in E$. The group of all automorphisms of $\mathcal{G}$ is denoted by $\mathrm{Aut}(\mathcal{G})$.

To eliminate these unnecessary computations, Ribeiro and Silva [108] introduce symmetry breaking conditions. The sub-graph represented by the left leaf of the g-trie in Figure 7.1 for example, has two automorphisms (see Figure 7.2), the identity map and $\varphi : u_1 \mapsto u_2, u_2 \mapsto u_1, u_3 \mapsto u_4, u_4 \mapsto u_3$. The bijection φ can be written in cyclic notation: $\varphi : (u_1u_2)(u_3u_4)$. If a node maps to itself it is omitted. The identity map in cyclic notation is $\varphi : (u_1)$.

The symmetry breaking condition $u_1 < u_2$ for the sub-graph depicted in Figure 7.2 will ensure that a candidate node that matches node u_2 has a higher index than the candidate node that matches u_1 and thus avoids counting the same sub-graph twice. Note that by fixing node u_2 all other nodes of the sub-graph are fixed.

$\varphi : (u_1)$ $\varphi : (u_1u_2)(u_3u_4)$

Figure 7.2: All automorphisms of the depicted sub-graph and their cyclic notations.

The G-tires algorithm was shown to outperform other motif detection algorithms (ESU [130, 131], Grochow [93], Kavosh [53]) in various tests [108].

7.2.2 QuateXelero

QuateXelero is another fast motif detection algorithm that is similar to G-tries [54]. The QuateXelero algorithm reduces computation time by minimising the number of times Nauty [73] is called. Nauty is a software that detects isomorphisms, produces canonical labellings and finds generators for a graph's group of automorphisms. Nauty is used in the G-tries algorithm [108] to produce the canonical labelling.

The data structure used by QuateXelero is slightly different to a g-trie. QuateXelero uses a quaternary tree, a tree where each node has at most four children. The quaternary tree structure is very useful to accommodate the enumeration of directed graphs. Figure 7.3 shows an example of a quaternary tree. Starting at the root of the tree, representing the first node of a sub-graph, the algorithm moves down the tree according to the way in which the second node is connected to the first. For example, if there is a directed edge pointing from the first node to the second node of the sub-graph, the algorithm follows the edge with label -1. The edge with label 0 corresponds to the two nodes not being connected, the edge with label 1 corresponds to a directed edge pointing from the second node to the first node and the edge with label 2 corresponds to an undirected edge between the two nodes (or two edges pointing in opposite directions). The next step down the tree corresponds to the connection between the third node of the sub-graph to the first, the following step down the tree corresponds to the connection between the third node and the second node, and so forth.

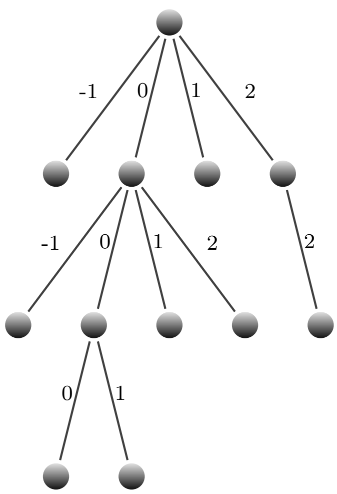


Figure 7.3: An example of a quaternary tree.

QuateXelero only calls Nauty when a particular sub-graph in the network is found. Khakabimamaghani et al. [54] argue that a large number of isomorphic sub-graphs will reach the same leaf of the quaternary tree and hence Nauty does not need to be called except for the first sub-graph that reaches the leaf. On the other hand, quaternary trees have greater order than g-tries as they do not exploit the common topologies of the different sub-graphs.

7.3 An algorithm for the bipartite clustering coefficients

In this section, we develop algorithms that count the sub-graphs needed for the calculation of the bipartite clustering coefficients (see Chapter 4). To build the algorithms we modify the G-tries algorithm. As we are enumerating undirected sub-graphs, we choose G-tries over QuateXelero. Furthermore, since we are interested in very particular sub-graphs, we can build the g-trie prior to running the algorithm and hence save the time needed to compute the groups of automorphisms and the canonical labellings.

7.3.1 Canonical labelling

To calculate the time dependent and time independent clustering coefficients, we need to enumerate all the sub-graphs depicted in Figure 7.4 and Figure 7.5 respectively. The canonical labelling of the individual sub-graphs is noted in the figures.

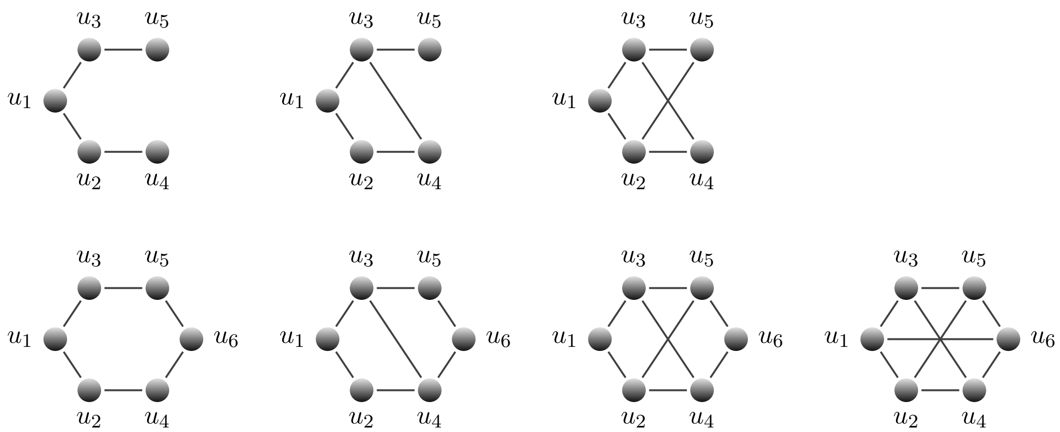


FIGURE 7.4: All sub-graphs, and their canonical labelling, that need to be enumerated to calculate the time dependent clustering coefficient.

From the canonical labelling it is now straight forward to build a g-trie that can store the different sub-graphs.

7.3.2 Building the g-tries

We build the g-trie depicted in Figure 7.6 to store and later enumerate the sub-graphs needed to calculate the time dependent clustering coefficient. Similarly Figure 7.7 depicts the g-trie that stores the sub-graphs needed to calculate the time independent clustering

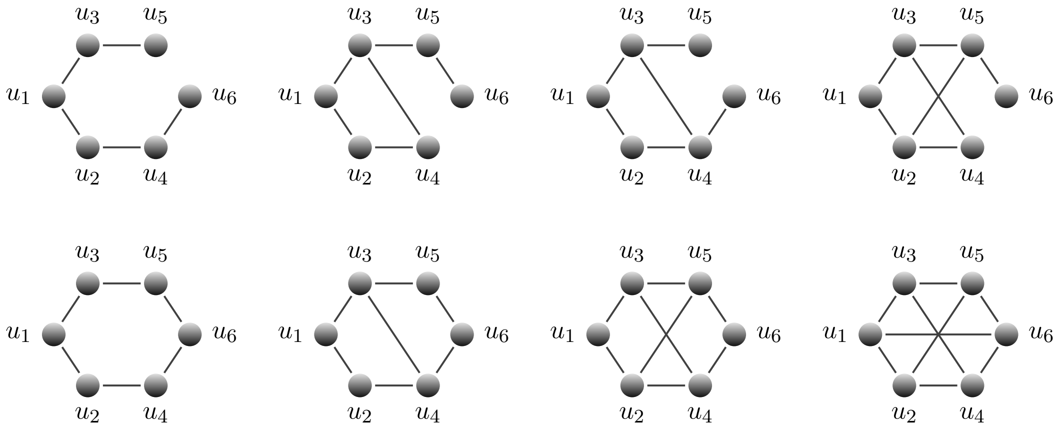


FIGURE 7.5: All sub-graphs, and their canonical labelling, that need to be enumerated to calculate the time independent clustering coefficient.

coefficients. Building the g-trie prior to enumerating the sub-graphs saves a large amount of computation time as Nauty [73] does not need to be called each time the algorithm is run.

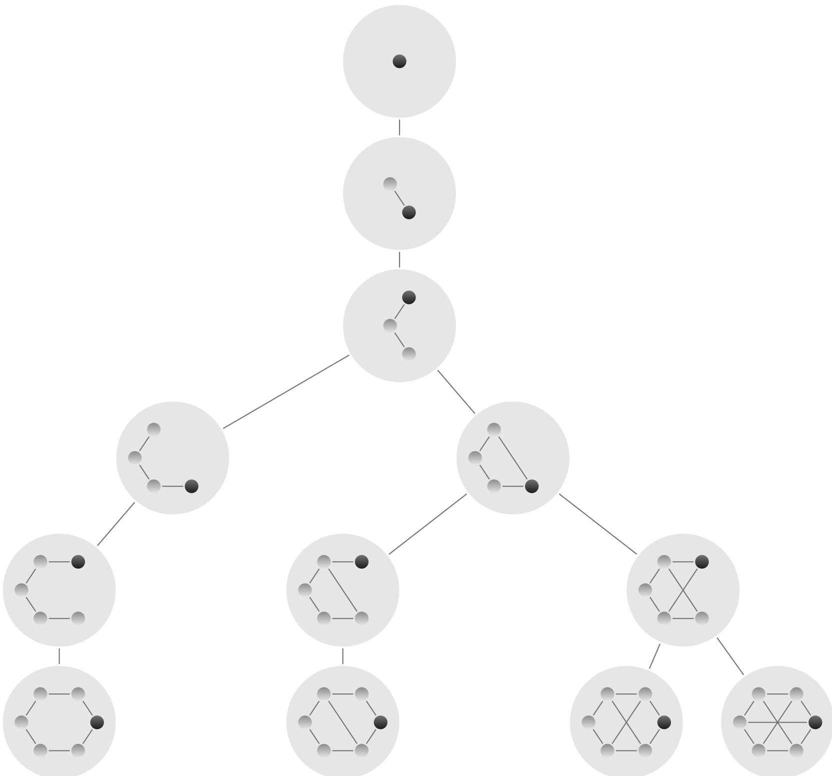


FIGURE 7.6: A g-trie that stores all the sub-graphs that need to be enumerated to measure the time dependent clustering coefficients

The main difference between the two g-tries depicted in Figure 7.6 and Figure 7.7 is that the g-trie depicted in Figure 7.6 stores the final sub-graphs (as depicted in Figure 7.4) in its leaves and the second last layer of the tree, whereas the g-trie depicted in Figure 7.7 stores the final sub-graphs only in its leaves.

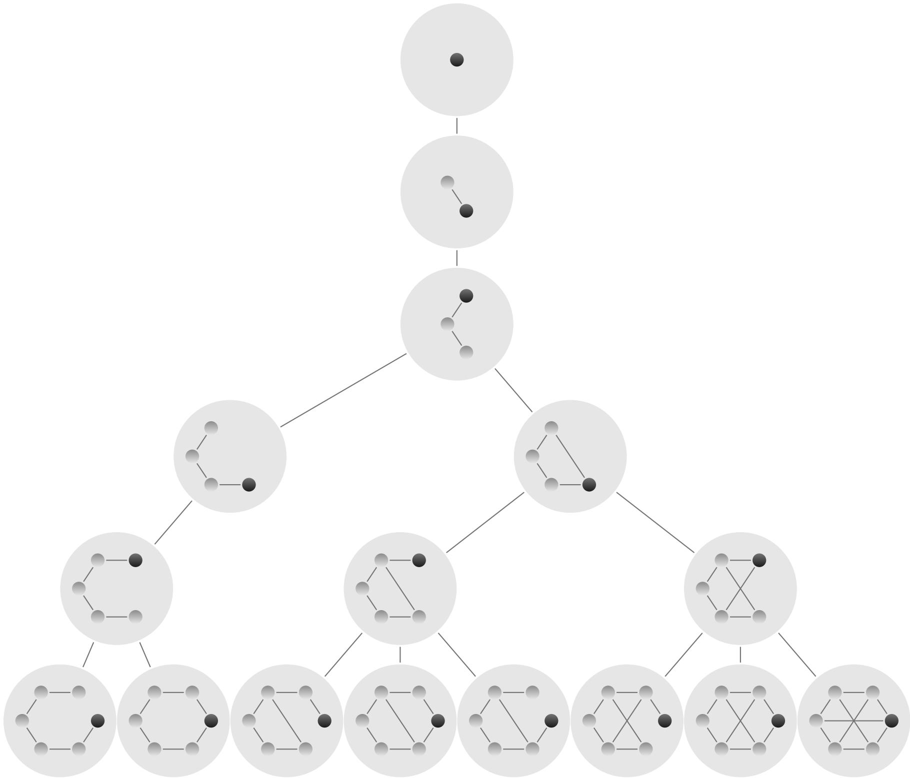


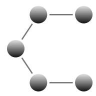
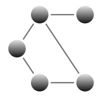
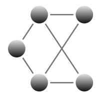
FIGURE 7.7: A g-trie that stores all the sub-graphs that need to be enumerated to measure the time independent clustering coefficients

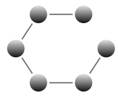
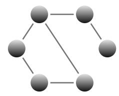
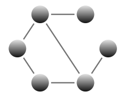
Finally, we add the symmetry breaking conditions to the g-trie to avoid multiple counting of the same sub-graph.

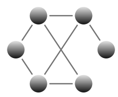
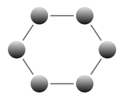
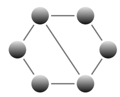
7.3.3 Symmetry breaking conditions

To find the symmetry breaking conditions we first find the group of automorphisms for each of the sub-graphs that we wish to enumerate. We have listed the groups in Table 7.1. Since some of the groups are very large, we have only listed the elements that generate the group, with all the elements in the groups being linear combinations of the elements that generate the group.

The symmetry breaking conditions can then be created such that all vertices of the sub-graphs are fixed. The breaking conditions are also listed in Table 7.1. For example, fixing node u_2 in the induced 4-path fixes all other vertices in that sub-graph.

$\mathcal{G}$			
Aut($\mathcal{G}$)	$< (u_2u_3)(u_4u_5) >$	$< (u_1u_4) >$	$< (u_1u_4),$ $(u_2u_3),$ $(u_4u_5) >$
Breaking conditions	$u_2 < u_3$	$u_1 < u_4$	$u_1 < u_4$ $u_2 < u_3$ $u_4 < u_5$

$\mathcal{G}$			
Aut($\mathcal{G}$)	$< (u_1u_2)(u_3u_4)(u_5u_6) >$	$< (u_1u_4) >$	$< (u_1u_2)(u_3u_4)(u_5u_6) >$
Breaking conditions	$u_1 < u_2$	$u_1 < u_4$	$u_1 < u_2$

$\mathcal{G}$			
Aut($\mathcal{G}$)	$< (u_1u_4),$ $(u_2u_3) >$	$< (u_2u_3)(u_4u_5),$ $(u_1u_2)(u_3u_4)(u_5u_6) >$	$< (u_1u_2)(u_3u_4)(u_5u_6),$ $(u_1u_5)(u_2u_6) >$
Breaking conditions	$u_1 < u_4$ $u_2 < u_3$	$u_1 < u_i \ \forall \ i = 2, \ldots, 6$ $u_2 < u_3$	$u_1 < u_2$ $u_1 < u_5$

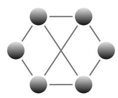
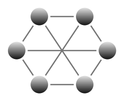
$\mathcal{G}$		
Aut($\mathcal{G}$)	$< (u_2u_3),$ $(u_4u_5),$ $(u_1u_6)(u_2u_4)(u_3u_5) >$	$< (u_2u_3),$ $(u_3u_6),$ $(u_4u_5),$ $(u_1u_2)(u_2u_4)(u_5u_6) >$
Breaking conditions	$u_1 < u_6$ $u_2 < u_3$ $u_4 < u_5$	$u_1 < u_i \ \forall \ i = 2, \ldots 6$ $u_2 < u_i \ \forall \ i = 3, \ldots 6$ $u_3 < u_i \ \forall \ i = 4, \ldots 6$ $u_4 < u_i \ \forall \ i = 5, \ldots 6$ $u_5 < u_6$

TABLE 7.1: The sub-graphs together with their groups of automorphisms and the symmetry breaking conditions. Since some of the automorphism groups are large, we have only listed the elements that generate the group.

After having identified all the necessary breaking conditions, we modify the G-tries algorithm to only count the sub-graphs of interest to us. Algorithm 2 displays the pseudo code that enumerates the sub-graphs that are required for the time independent clustering coefficient. Since the algorithm is very large, some of the code has been omitted.

Algorithm 2 Enumerating sub-graphs required for the time independent clustering coefficient.

```
procedure EnumerateInependent(𝒢 = (U, V))
  for u ← 1, |U| do
    visited = {u}
    N ← {n_i | (u, n_i) ∈ E, ∀ i}                                   ▷ Level one of g-trie
    for i ← 1, |N| do
      visited = n_i ∪ visited
      for j ← 1, |N| do
        if n_j ∉ visited then
          visited = n_j ∪ visited
          NN ← {nn_k | (n_i, nn_k) ∈ E, ∀ k}                         ▷ Level three of g-trie
          for k ← 1, |NN| do
            if nn_k ∉ visited ∧ (nn_k, n_j) ∉ E ∧ u < n_i then
              visited = nn_k ∪ visited
              NNN ← {nnn_l | (n_j, nnn_l) ∈ E, ∀ l}                  ▷ Level four of g-trie
              for l ← 1, |NNN| do
                if nnn_l ∉ visited ∧ (nnn_l, n_i) ∉ E then
                  visited = nnn_l ∪ visited
                  NNNN ← {nnnn_m | (nn_k, nnnn_m) ∈ E, ∀ m}          ▷ Level five
                  for m ← 1, |NNNN| do
                    if nnnn_m ∉ visited ∧ (nnnn_m, u) ∉ E ∧ (nnnn_m, nnn_l) ∉ E then
                      induced4Path ← induced4Path + 1
                    end if
                    if nnnn_m ∉ visited ∧ (nnnn_m, u) ∉ E ∧ (nnnn_m, nnn_l) ∈ E ∧ u < n_j, nn_k, nnn_l, nnnn_m ∧ n_i < n_j then
                      induced6Cycle ← induced6Cycle + 1
                    end if
                  end for
                end if
                visited = visited \ nnn_l
              end for
            end if
            if (nn_k, n_i) ∈ E∧  then
              . . .
            end if
            visited = visited \ nn_k
          end for
        end if
        visited = visited \ n_j
      end for
      visited = visited \ n_i
    end for
    visited = visited \ u
  end for
end procedure
```

We have written the pseudo code, following down the g-trie to two of its leaves. The omitted code follows the same principle. The algorithm that enumerates the sub-graphs for the time dependent clustering coefficient works in the exact same way. In fact, we could simply place the counters in the appropriate places of Algorithm 2 to enumerate the required sub-graphs.

7.3.4 Discussion

This section demonstrated the building of g-tries for the enumeration of the sub-graphs needed for the calculation of the time dependent and time independent clustering coefficients. The primary motivation for altering the G-tries algorithm is that it is one of the fastest algorithms for sub-graph enumeration. Since we are only interested in very particular sub-graphs and not all sub-graphs of a particular size, we were able to further speed up the enumeration. Another reason for writing the G-tries algorithm with some modifications ourselves is that, although the algorithm is publicly available, we were unable to install it to run properly on the computer.

7.4 Enumerating paths on the square lattice

The enumeration of paths, also called self avoiding walks, is a very interesting as well as challenging problem. It is not only of theoretical interest, but finds many applications. Some example include protein folding [19] and the design of telephone networks [46]. The motion of particles is also modelled by lattice paths [51]. As Guttmann [46] points out, the literature contains a huge number of numerical results and very few formal proofs.

Guttmann [46] describes the problem very aptly by calling it simple in its definition, yet extremely challenging to solve. This section discusses new ways to solve the problem of path enumeration on the square lattice. We believe it is worth considering this problem since the square lattice is a regular bipartite network and results would lead towards the understanding of other, non-regular, bipartite networks.

Instead of tackling the main problem of enumerating all paths of length n, researchers have tackled smaller sub-problems that are easier to solve (see [51] and the references therein). The problem can be simplified by, for instance considering only one quadrant of the square lattice or by limiting the directions the path is allowed to take.

7.4.1 Definitions

Since lattice path enumeration has many applications in different areas, notation often varies. We use the notation given in this subsection.

Definition 7.2. The two dimensional *square lattice* $\mathcal{L}$ is represented by the Cayley graph of the additive group $\mathbb{Z}^2$.

Drawing the square lattice in the Cartesian plane enables the assignment of a unique position, denoted by the ordered pair (x, y), to each node.

Figure 7.8 shows the two dimensional square lattice, with the origin coloured in light grey.

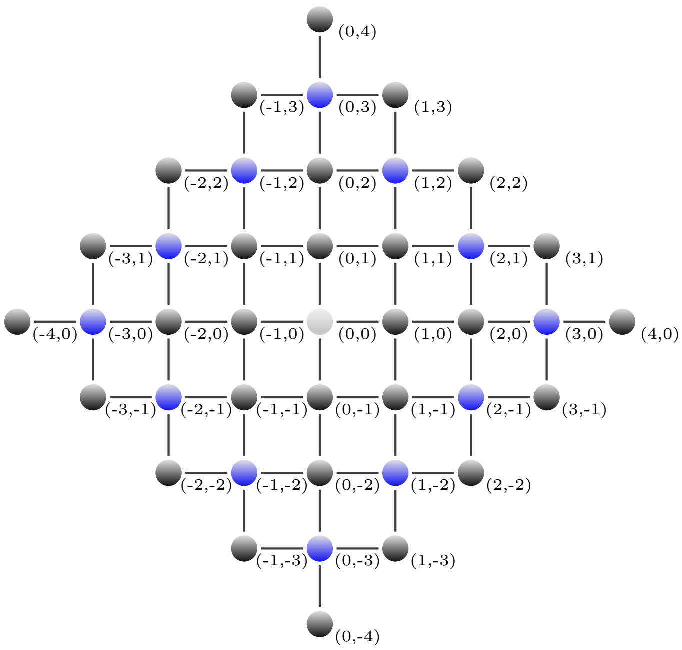


FIGURE 7.8: The two dimensional square lattice showing the coordinates of the individual nodes. The light grey node in the centre of the lattice is the origin. The blue nodes form the third layer of the lattice.

A path of length n (see Definition 2.13) is denoted by P_n. The number of paths of length n is denoted by $N(P_n)$.

Definition 7.3. The *step set* S is a fixed and finite set of possible steps that a path is allowed to take on a given lattice.

When counting the number of all n-paths on the square lattice, without any restrictions, the step set is given by $S = \{(1,0),(0,1),(-1,0),(0,-1)\}$. Step sets are often denoted using the directions *north*, *south*, *east* and *west*. Henceforth, for simplicity, we use the latter notation for step sets and instead write $S = \{N, E, W, S\}$.

Definition 7.4. The n^{th} *layer* of the square lattice $\mathcal{L}$, contains all nodes that are exactly n steps away from the origin.

The distance or the number of steps that a node of the square lattice is away from the origin can be calculated simply by adding up the absolute values of its x and y-coordinates.

In what follows, we make frequent use of the binomial coefficient. The binomial coefficient $\begin{pmatrix} n \\ k \end{pmatrix}$, read n choose k, is defined as

$$\begin{pmatrix} n \\ k \end{pmatrix} = \frac{n!}{k!(n-k)!}, \tag{7.1}$$

where $0 \leq k \leq n$. If $n < k$, the binomial coefficient is equal to zero by definition.

A concise review of the work on lattice path enumeration is presented in [51]. The majority of the literature concentrates on restricted paths that use a two element step set, that is the path can take only two different directions. A path is restricted if it is limited by boundaries.

7.4.2 Pascal's triangle

Pascal's triangle is depicted in Figure 7.9. Each row of Pascal's triangle corresponds to the coefficients of the binomial expansion of $(a+b)^n$, where n is the index of the row. Laying Pascal's triangle over the first quadrant (for example), of the square lattice, gives the number of paths that start at the origin and end at the corresponding node, with the path having length equal to the distance of the end node to the origin.

Proposition 7.5. *Suppose that we restrict the path P_n to the first quadrant of the square lattice and consider the step set $S = \{N, E\}$. Then the number of paths of length n that*

$$
\begin{array}{ccccccccccccccc}
 & & & & & & & 1 & & & & & & & \\
 & & & & & & 1 & & 1 & & & & & & \\
 & & & & & 1 & & 2 & & 1 & & & & & \\
 & & & & 1 & & 3 & & 3 & & 1 & & & & \\
 & & & 1 & & 4 & & 6 & & 4 & & 1 & & & \\
 & & 1 & & 5 & & 10 & & 10 & & 5 & & 1 & & \\
 & 1 & & 6 & & 15 & & 20 & & 15 & & 6 & & 1 & \\
1 & & 7 & & 21 & & 35 & & 35 & & 21 & & 7 & & 1
\end{array}
$$

FIGURE 7.9: Pascal's triangle

start at the origin and end at the node at position (i, j), where $|i| + |j| = n$ is given by $\begin{pmatrix} n \\ j \end{pmatrix}$.

Proof. Without loss of generality, consider the first quadrant of the square lattice $\mathcal{L}$. Since $|i|+|j| = n$, the node at position (i, j) is located at distance n from the origin and can be written as $(n-j, j)$, where $j \in \{0, \ldots, n\}$. Hence, a path of length n from the origin to node (i, j) is a sequence of $(n-j)$ Es and j Ns. There are $\begin{pmatrix} n \\ j \end{pmatrix}$ different combinations of positions for the Ns in the sequence. □

Corollary 7.6. *The number of paths of length n ending in the n^{th} layer of the square lattice is given by* $4 \sum_{i=0}^{n-1} \begin{pmatrix} n \\ i \end{pmatrix} = 4(2^n - 1)$.

7.4.3 Preliminary results

In this subsection, we present preliminary results, give remarks and state observations related to the problem of enumerating all paths of length n on the square lattice $\mathcal{L}$. We deduce formulæ for calculating the number of n-paths that start at the origin and end in the i^{th} layer, where $i < n$. This is useful because if, say n is even, the number of paths of length n, starting at the origin of the square lattice, is the sum of the number of n-paths that end in all even layers less than or equal to n.

Note that there are $4n$ nodes in the n^{th} layer of the square lattice. There are $(n+1)$ nodes in the n^{th} layer of each of the four quadrants of the square lattice since the nodes at positions $(0, n), (n, 0), (0, -n)$ and $(-n, 0)$ are each located in two quadrants.

Proposition 7.7. *The number of paths of length n, where $n \geq 3$, starting at the origin and ending in the $(n-2)^{th}$ layer of the square lattice is given by*

$$\begin{aligned} N(P_n)_{(n-2)} &= 4\sum_{k=0}^{n-3}\left[2\left[\binom{n-2}{k+1}+\binom{n-2}{k-1}\right]\right. \\ &\qquad \left. +(n-2)\left[\binom{n-3}{k+1}+\binom{n-3}{k-2}\right]\right] \end{aligned} \tag{7.2}$$

Proof. Without loss of generality, consider the first quadrant of the square lattice $\mathcal{L}$. Any node in the first quadrant located in the $(n-2)^{\text{th}}$ layer can be written as $(n-k-2, k)$, where $k \in \{0, \ldots, (n-2)\}$. The path $P_{(n-2)}$, starting at the origin and ending at node $(n-k-2, k)$ is a sequence of steps N and E. Hence,

$$P_{(n-2)} = \underbrace{NN\ldots N}_{k\ N\text{s}}\ \underbrace{EE\ldots E}_{(n-k-2)\ E\text{s}}\ ,$$

or any permutation of this sequence.

The path P_n that ends at node $(n-k-2, k)$ is a continuation of $P_{(n-2)}$ with two additional steps, with the second step being an inverse of the first. There are exactly two possibilities NS and EW.

By definition the elements of a path are unique (see Definition 2.13) and hence,

i) If $S(W)$ is the first step of the path, the next step has to be E (N respectively).

ii) If $S(W)$ is the last step of the path, it has to be preceded by the step E (N respectively).

iii) If $S(W)$ is neither the first nor the last step of the path it has to be preceded and followed by the step E (N respectively).

It follows that $N(P_n)_{(n-2)}$ is given by the number of valid permutations of the string $\underbrace{NN\ldots N}_{k\ N\text{s}}\ \underbrace{EE\ldots E}_{(n-k-2)\ E\text{s}}\ NS$, restricted by the above constraints plus the number of valid permutations of the string $\underbrace{NN\ldots N}_{k\ Ns}\ \underbrace{EE\ldots E}_{(n-k-2)\ Es}\ EW$, restricted by the above constraints.

If S is the first or the last step of the path, then the remaining steps can be arranged in $\begin{pmatrix} n-2 \\ k+1 \end{pmatrix}$ possible ways.

If S is neither the start nor the end of the path, it may be placed in $(n-2)$ different positions. The remaining steps can be arranged in $\begin{pmatrix} n-3 \\ k+1 \end{pmatrix}$ possible ways.

If W is the first or the last step of the path, then the remaining steps can be arranged in $\begin{pmatrix} n-2 \\ k-1 \end{pmatrix}$ possible ways.

If W is neither the start nor the end of the path, it may be placed in $(n-2)$ different positions. The remaining steps can be arranged in $\begin{pmatrix} n-3 \\ k-2 \end{pmatrix}$ possible ways.

Since the four quadrants of the square lattice are isomorphic to each other, the total number of n-paths starting at the origin and ending in the $(n-2)^{\text{th}}$ layer is given by

$$\begin{aligned} N(P_n)_{(n-2)} &= 4\sum_{k=0}^{n-3}\left[2\left[\begin{pmatrix} n-2 \\ k+1 \end{pmatrix}+\begin{pmatrix} n-2 \\ k-1 \end{pmatrix}\right]\right. \\ &\left. +(n-2)\left[\begin{pmatrix} n-3 \\ k+1 \end{pmatrix}+\begin{pmatrix} n-3 \\ k-2 \end{pmatrix}\right]\right] \end{aligned}$$

□

We can proceed in the same manner to find the number of paths of length n from the origin to the $(n-4)^{\text{th}}$ layer and so on.

Proposition 7.8. *The number of paths of length n, where $n \geq 5$, starting at the origin and ending in the $(n-4)^{th}$ layer of the square lattice is given by*

$$N(P_n)_{(n-4)} = 4\sum_{k=0}^{n-5}\left[2\left[\begin{pmatrix} n-3 \\ k+2 \end{pmatrix}+\begin{pmatrix} n-3 \\ k-1 \end{pmatrix}+\begin{pmatrix} n-4 \\ k-1 \end{pmatrix}+\begin{pmatrix} n-4 \\ k+1 \end{pmatrix}\right.\right.$$

$$
\begin{aligned}
&+\binom{n-4}{k}+(n-5)\binom{n-5}{k-1}+(n-5)\binom{n-5}{k}\Bigg]\\
&+3\left[\binom{n-4}{k+2}+\binom{n-4}{k-2}+(n-4)\binom{n-5}{k+2}\right.\\
&+(n-4)\binom{n-5}{k-3}\Bigg]+(n-3)\left[\binom{n-4}{k+2}\right.\\
&+\binom{n-4}{k-2}\Bigg]+\binom{n-4}{2}\left[\binom{n-6}{k+2}+\binom{n-6}{k-4}\right]\\
&+\binom{n-5}{2}\binom{n-6}{k-1}\Bigg]
\end{aligned}
\tag{7.3}
$$

Proof. Without loss of generality, consider the first quadrant of the square lattice $\mathcal{L}$. Any node in the first quadrant located in the $(n-4)^{\text{th}}$ layer can be written as $(n-k-4, k)$, where $k \in \{0, \ldots, (n-4)\}$. The path $P_{(n-4)}$ starting at the origin and ending at node $(n-k-4, k)$ is a sequence of steps N and E. Hence,

$$
P_{(n-4)} = \underbrace{NN\ldots N}_{k\ Ns}\ \underbrace{EE\ldots E}_{(n-k-4)\ Es},
$$

or any permutation of this sequence.

The path P_n that ends at node $(n-k-4, k)$ is a continuation of $P_{(n-4)}$ with four additional steps. To reach the $(n-4)^{\text{th}}$ layer, we need two pairs of steps that are inverse of each other. There are exactly three possibilities $NNSS$, $EEWW$ and $NSEW$.

If the additional steps are $NNSS$, the two S steps can be placed in the following positions:

i) The two S steps can be placed in the first two places of the sequence and have to be followed by an E step. The remaining steps can be arranged in $\binom{n-3}{k+2}$ possible ways. Similarly, the two S steps can be placed in the last two places of the sequence.

ii) One S can be placed in the first position of the sequence, followed by an E and one S can be placed in the last position of the sequence, preceded by an E. The remaining steps can be arranged in $\binom{n-4}{k+2}$ possible ways.

iii) One S can be placed in the first position of the sequence, followed by an E and the second S can be placed anywhere except the last or the third position. The second S is preceded and followed by an E. The remaining steps can be arranged in $(n-4)\binom{n-5}{k+2}$ possible ways. Similarly, the first S can be placed in the last position.

iv) One S is placed in the first position of the sequence, followed by an E, the second S is placed in the third position of the sequence, followed by an E. The remaining steps can be arranged in $\binom{n-4}{k+2}$ possible ways. Similarly, the first S can be placed in the last position and the second S in the third last position.

v) The two S steps can be placed together anywhere in the sequence, except at the start or the end. The two S steps are preceded and followed by an E. The remaining steps can be arranged in $(n-3)\binom{n-4}{k+2}$ possible ways.

vi) The first S step is placed anywhere in the sequence, preceded by an E and followed by an E, followed by the second S step that is followed by an E. The remaining steps can be arranged in $(n-4)\binom{n-5}{k+2}$ possible ways.

vii) The two S steps can be placed anywhere in the sequence, both followed and preceded by E steps. The remaining steps can be arranged in $\binom{n-4}{2}\binom{n-6}{k+2}$ possible ways.

If the additional steps are $EEWW$, the two W steps can be placed in the following positions:

i) The two W steps can be placed in the first two places of the sequence and have to be followed by an N step. The remaining steps can be arranged in $\binom{n-3}{k-1}$

possible ways. Similarly, the two W steps can be placed in the last two places of the sequence.

ii) One W can be placed in the first position of the sequence, followed by an N and one W can be placed in the last position of the sequence, preceded by an N. The remaining steps can be arranged in $\binom{n-4}{k-2}$ possible ways.

iii) One W can be placed in the first position of the sequence, followed by an N and the second W can be placed anywhere except the last or the third position. The second W is preceded and followed by an N. The remaining steps can be arranged in $(n-4)\binom{n-5}{k-3}$ possible ways. Similarly, the first W can be placed in the last position.

iv) One W is placed in the first position of the sequence, followed by an N, the second W is placed in the third position of the sequence, followed by an N. The remaining steps can be arranged in $\binom{n-4}{k-2}$ possible ways. Similarly, the first W can be placed in the last position and the second W in the third last position.

v) The two W steps can be placed together anywhere in the sequence, except at the start or the end. The two W steps are preceded and followed by an N. The remaining steps can be arranged in $(n-3)\binom{n-4}{k-2}$ possible ways.

vi) The first W step is placed anywhere in the sequence, preceded by an N and followed by an N, followed by the second W step that is followed by an N. The remaining steps can be arranged in $(n-4)\binom{n-5}{k-3}$ possible ways.

vii) The two W steps can be placed anywhere in the sequence, both followed and preceded by N steps. The remaining steps can be arranged in $\binom{n-4}{2}\binom{n-6}{k-4}$ possible ways.

If the additional steps are $NSEW$, the S step and the W step can be placed in the following positions:

i) The S step can be placed in the first position of the sequence followed by the W step. The third and forth positions of the sequence have to be N steps. The remaining steps can be arranged in $\begin{pmatrix} n-4 \\ k-1 \end{pmatrix}$ possible ways. Similarly the S step can be placed in the last position of the sequence, proceeded by the W step.

ii) The W step can be placed in the first position of the sequence followed by the S step. The third and forth positions of the sequence have to be E steps. The remaining steps can be arranged in $\begin{pmatrix} n-4 \\ k+1 \end{pmatrix}$ possible ways. Similarly the W step can be placed in the last position of the sequence, proceeded by the S step.

iii) The S step can be placed in the first position of the sequence, followed by an E. The W step can be placed in the last position of the sequence, preceded by an N. The remaining steps can be arranged in $\begin{pmatrix} n-4 \\ k \end{pmatrix}$ possible ways. Similarly the W step can be placed in the first position of the sequence and the S step can be placed in the last position of the sequence.

iv) The S step is placed in the first position of the sequence, followed by an E. The W step is placed anywhere in the sequence, except in the third or the last position of the sequence and is followed and preceded by N steps. The remaining steps can be arranged in $(n-5)\begin{pmatrix} n-5 \\ k-1 \end{pmatrix}$ possible ways. Similarly, the S step can be placed in the last position of the sequence.

v) The W step is placed in the first position of the sequence, followed by an N. The S step is placed anywhere in the sequence, except in the third or the last position of the sequence and is followed and preceded by E steps. The remaining steps can be arranged in $(n-5)\begin{pmatrix} n-5 \\ k \end{pmatrix}$ possible ways. Similarly, the W step can be placed in the last position of the sequence.

vi) The S step can be placed anywhere in the sequence, except the first and the last position and is followed and preceded by E steps. The W step can be placed anywhere in the sequence, except the first and the last position and is followed and preceded by N steps. The remaining steps can be arranged in $\begin{pmatrix} n-5 \\ 2 \end{pmatrix}\begin{pmatrix} n-6 \\ k-1 \end{pmatrix}$ possible ways.

Since the four quadrants of the square lattice are isomorphic to each other, the total number of n-paths starting at the origin and ending in the $(n-4)^{\text{th}}$ layer is given by

$$\begin{aligned}
N(P_n)_{(n-4)} &= 4\sum_{k=0}^{n-5}\Bigg[2\Bigg[\binom{n-3}{k+2}+\binom{n-3}{k-1}+\binom{n-4}{k-1}+\binom{n-4}{k+1}\\
&\quad+\binom{n-4}{k}+(n-5)\binom{n-5}{k-1}+(n-5)\binom{n-5}{k}\Bigg]\\
&\quad+3\Bigg[\binom{n-4}{k+2}+\binom{n-4}{k-2}+(n-4)\binom{n-5}{k+2}\\
&\quad+(n-4)\binom{n-5}{k-3}\Bigg]+(n-3)\Bigg[\binom{n-4}{k+2}\\
&\quad+\binom{n-4}{k-2}\Bigg]+\binom{n-4}{2}\Bigg[\binom{n-6}{k+2}+\binom{n-6}{k-4}\Bigg]\\
&\quad+\binom{n-5}{2}\binom{n-6}{k-1}\Bigg]
\end{aligned}$$

□

Continuing to calculate all paths of length n that start at the origin of the square lattice and end in a given layer becomes quickly infeasible to do manually. For instance, the next step would be to find the number of paths of length n from the origin of the square lattice to the $(n-6)^{\text{th}}$ layer. We know that the path P_n that ends at node $(n-k-6,k)$ is a continuation of $P_{(n-6)}$ with six additional steps. To reach the $(n-6)^{\text{th}}$ layer, we need three pairs of steps that are inverse of each other. There are exactly four possibilities $NNNSSS$, $EEEWWW$, $NNSSEW$ and $NSEEWW$.

The number of ways in which the additional steps can be placed in the sequence of steps becomes large very quickly. In addition, there are more restrictions on the placement of steps. Not only can a step not be followed or proceeded by its inverse, but for instance the sub-sequence $WNES$ is not allowed to occur, since it forms a cycle of length four.

While studying the number of paths from the origin to a given layer, we made the following observations.

Conjecture 7.9. *The sum of the $(n-2)^{th}$ row of the triangle depicted in Figure 7.10 gives the number of paths of length n from the origin to nodes in the $(n-2)^{th}$ layer, located in the first quadrant of the square lattice.*

$$
\begin{array}{c}
0 \ {\scriptstyle +2} \\
2 \ {\scriptstyle +0} \ 2 \ {\scriptstyle +4} \\
6 \ {\scriptstyle +1} \ 4 \ {\scriptstyle +1} \ 6 \ {\scriptstyle +6} \\
12 \ {\scriptstyle +3} \ 11 \ {\scriptstyle +2} \ 11 \ {\scriptstyle +3} \ 12 \ {\scriptstyle +8} \\
20 \ {\scriptstyle +6} \ 26 \ {\scriptstyle +5} \ 24 \ {\scriptstyle +5} \ 26 \ {\scriptstyle +6} \ 20 \ {\scriptstyle +10} \\
30 \ {\scriptstyle +12} \ 52 \ {\scriptstyle +11} \ 55 \ {\scriptstyle +10} \ 55 \ {\scriptstyle +11} \ 52 \ {\scriptstyle +12} \ 30 \ {\scriptstyle +12} \\
42 \ {\scriptstyle +24} \ 94 \ {\scriptstyle +23} \ 118 \ {\scriptstyle +21} \ 120 \ {\scriptstyle +21} \ 118 \ {\scriptstyle +23} \ 94 \ {\scriptstyle +24} \ 42
\end{array}
$$

FIGURE 7.10: The sum of the $(n-2)^{\text{th}}$ row of the triangle gives the number of paths of length n from the origin to nodes in the $(n-2)^{\text{th}}$ layer, located in the first quadrant of the square lattice. The tip of the triangle is row zero. The triangle is generated in a similar manner as Pascal's triangle, with the difference being the red numbers are added to the element in the next row. The red numbers on jump are generated by adding the red numbers of the previous row. The first and last red number in each row are generated by adding one to the second number in the same row.

7.4.4 Discussion

This subsection presented a brief overview of the well known problem of the enumeration of paths on the square lattice. Proving Propositions 7.7 and 7.8 demonstrated the complexity of the problem. The proof of Conjecture 7.9 is left for future work.

7.5 Summary

This chapter was dedicated to the enumeration of sub-graphs. We described two existing algorithms for the detection of motifs that we used to enumerate the sub-graphs needed to calculate the time dependent and time independent clustering coefficients (see Chapter 4).

Next, we focused on a very different but related problem, that of enumerating the paths on the square lattice. After a brief overview of the literature, we presented some preliminary work, demonstrating the complexity of the problem. The enumeration of all paths of length n on the square lattice is an open problem that presents much scope for future work. We believe that a solution to this problem is a first steps towards understanding how to efficiently enumerate paths in irregular bipartite networks.

Chapter 8

Conclusion

Networks are a useful means of representing and studying real world phenomena. In this thesis we examined bipartite networks with the aim of uncovering significant behaviour. Bipartite networks are particular types of networks, comprising of two different sets of nodes and are far less studied than ordinary networks. We approached the analysis of bipartite networks in two different ways. First, by overcoming some of the limitations of one-mode projections and, second, by defining new network measures particularly suited to bipartite networks.

8.1 Contributions

In Chapter 3 we developed a novel technique to extract the backbone of one-mode projections. Extracting the backbone of a network aims to reduce the amount of redundant information in networks. It is important to note that edges with high weight are not necessarily the most significant edges in the network. If the network is a one-mode projection, the identification of significant edges is even more challenging as weights in the projection depend on the degree distributions of the bipartite network.

We were able to prove that the edge weights of a one-mode projection of a bipartite network follow a Poisson binomial distribution. This result enabled us to efficiently determine the significance of individual connections in the projection of a bipartite network. Being able to calculate the weight distribution in the projection from the degree distributions of the bipartite network saves valuable computation time. Previously, the weights of projections had to be compared to the weights of the projections of an ensemble of random bipartite networks in order to determine their significance. The time consuming process of projecting several hundreds of bipartite networks has been made unnecessary by our technique of extracting the backbone.

Extracting the backbone of several real world networks demonstrated that many insignificant edges were found to link nodes of different communities of the projection. Hence, backbone extraction is an aid in the detection of network communities.

In contrast to Chapter 3, Chapters 4 and 5 pursued the direct analysis of bipartite networks.

In Chapter 4 we developed bipartite clustering coefficients for the analysis of bipartite networks. Our clustering coefficients are designed to solve the limitations of previously proposed clustering coefficients for bipartite networks. Unlike many other bipartite clustering coefficients, we followed the path taken by Opsahl [94] and measured triadic closure of the nodes in a bipartite network. In other words, we were interested in how well any three nodes of the same type were connected to each other. Since the way in which bipartite clusters form over time directly depends on the type of bipartite network, we identified two different types of bipartite networks, time dependent and time independent networks, that develop very differently over time. This difference in formation has been ignored in the literature thus far. Another limitation, that of information loss, is overcome by our clustering coefficients by distinguishing between differently structured clusters. We introduced clustering coefficients for time dependent and time independent bipartite networks as well as for the differently structured clusters.

Chapter 5 was dedicated to the applications of our clustering coefficients. We demonstrated that it may be used as a tool to identify the most influential nodes in a bipartite network as well as for the prediction of the future popularity of new items in rating networks.

Intuitively it is felt that the most influential nodes of a network would have a high degree. However, this is usually not the case [65] and more sophisticated techniques for ranking the nodes of a network by importance were needed. The clustering coefficients we defined show great potential in identifying important nodes in one-mode networks.

By introducing the driving score measure, we were able to use the clustering coefficients to identify the most influential nodes of two real world networks.

Next, we used our bipartite clustering coefficients to predict the future popularity of new items in rating networks. Predicting the popularity of a new item is particularly hard, due to the lack of information about that item. By analysing the clustering behaviour of the first user who rated a new item we were able to considerably improve current prediction rates. We demonstrated our approach on the MovieLens network and the Digg network getting in the first case an improvement to approximately 65% and in the second case an improvement to approximately 50%.

In Chapter 6 we combined backbone extraction and bipartite clustering coefficients in the analysis of crime networks. We presented two case studies, one of data that was collected in the state of New South Wales, Australia and the other of data collected in the United Kingdom. While the work presented in Chapter 6 is work in progress, we demonstrated that motifs previously identified in a burglary network are likely to be the result of projecting the network. We raised many questions in Chapter 6 that need to answered in future work.

Chapter 7 described two motif detection algorithms, one of which we modified for the enumeration of the sub-graphs that are needed to calculate the bipartite clustering coefficients in time dependent and time independent networks. The second part of Chapter 7 presented preliminary work on the enumeration of paths on the square lattice.

8.2 Future work

This research has in many way given rise to many more questions than have been answered in this thesis. In this section we give a list of topics that we plan to investigate in future research.

8.2.1 Generating random bipartite networks

In Chapter 2, Subsection 2.2.5.3 we described an algorithm for generating random bipartite networks. The algorithm requires a pair of bipartite graphic sequences as its input. The study of graphic and bipartite graphic sequences is an active area of research and it would be worth exploring whether there are efficient ways of finding pairs of bipartite graphic sequences that can be used for creating random complex networks.

8.2.2 Finding non-isomorphic matrices with non-negative entries

In Chapter 3, Subsection 3.2.1.1 we described the method given in [33] to construct two non-isomorphic matrices that have identical primary and secondary projections. We pointed out that one matrix is simply the negation of the other. Since many real world networks have positive edge weights, we are interested in finding other ways of constructing non-isomorphic matrices with non-negative entries.

8.2.3 Extracting the backbone of bipartite networks

The aim of Chapter 3 is not the partitioning of the original bipartite network, but to speed up the extraction of the backbone of one-mode projections. Whether it is possible to directly extract the backbone of bipartite networks in a similar manner will be investigated in future research.

8.2.4 Performance of community detection algorithms

The community analysis in Chapter 3 is purely instrumental in assessing the backbone extraction method. The obvious next step would be to assess the performance of different community detection algorithms on backboned projections. Since it is not clear how to compare community detection algorithms that are based on different approaches it is necessary to firstly find methods of comparison and is left to future work.

8.2.5 An expression for the clustering coefficient of projections from random bipartite networks

In Chapter 4 we formally showed that the clustering coefficient of projections is generally higher than the clustering coefficient of random networks with the same degree distribution. The next step is to find an expression that approximates the clustering coefficient of projections of random bipartite networks. Another interesting task related to this is finding the weight distribution of binary one-mode projections. Both of these are left for future work.

8.2.6 Implementation of recommendation systems

Chapter 5 presented the applications of the bipartite clustering coefficients. By analysing the clustering behaviour of users in rating networks, we were able to improve the predictions of the future popularity of new items in rating networks. In future work, we would look to implement our prediction method into a real time recommendation system.

8.2.7 Improving popularity predictions by considering more than one user

The predictions of future popularity in Chapter 5 were made by considering only a single user. We are curious whether predictions can be improved by considering more than one user. While analysing information about other users may improve predictions about an item, one needs first to be able to handle the larger amount of data that will increase the already large computation time. We also plan to predict the, say 100, most popular items in the near future.

8.2.8 Crime networks

Chapter 6 presents ample scope for future work. The analyses carried out in Chapter 6 gave rise to many questions that need to be addressed in future work. The study of the New South Wales crime dataset in Chapter 6 demonstrated the analytical power of the techniques for the analysis of bipartite networks that we developed in this thesis, while simultaneously raising many questions giving opportunities for future work. We believe that further analysis of crime data using complex bipartite networks would advance the understanding of criminal activity and thus lead to the implementation of more efficient prevention measures.

8.2.9 Path enumeration

The problem of enumerating all paths of length n on the square lattice is a well known open problem in mathematics and presents much scope for future work. We hope to continue working on this problem in coming research, starting by finding a proof for Conjecture 7.9.

Appendix A

Figures

A.1 Significant connections in the MovieLens tag genome network

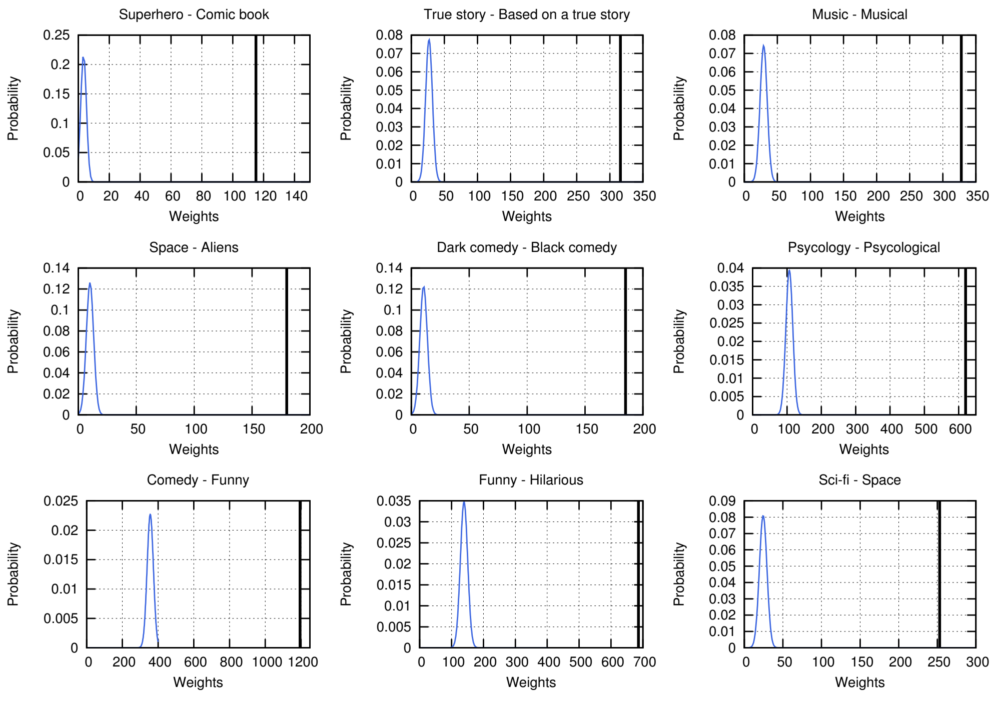


FIGURE A.1: The weight probability distributions of the nine most significant edges in the tag-tag projection, where the observed weight is greater than expected. The blue curves show the approximated probability distributions, the black vertical bars mark the observed weight in the weighted one-mode projection of the MovieLens Tag Genome network (100 most popular tags).

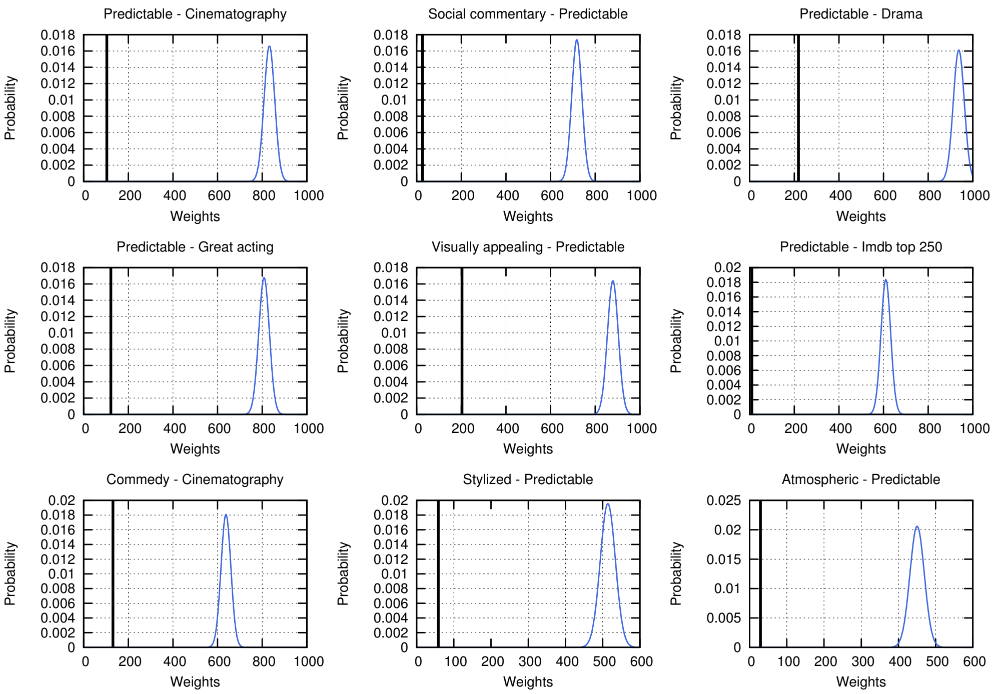


FIGURE A.2: The weight probability distributions of nine edges in the tag-tag projection, where the observed weight is smaller than expected (not included in the backbone). These edges represent antagonisms. The blue curves show the approximated probability distributions, the black vertical bars mark the observed weight in the weighted one-mode projection of the MovieLens Tag Genome network (100 most popular tags).

A.2 The neighbourhood of senators of the 108th U.S. Senate

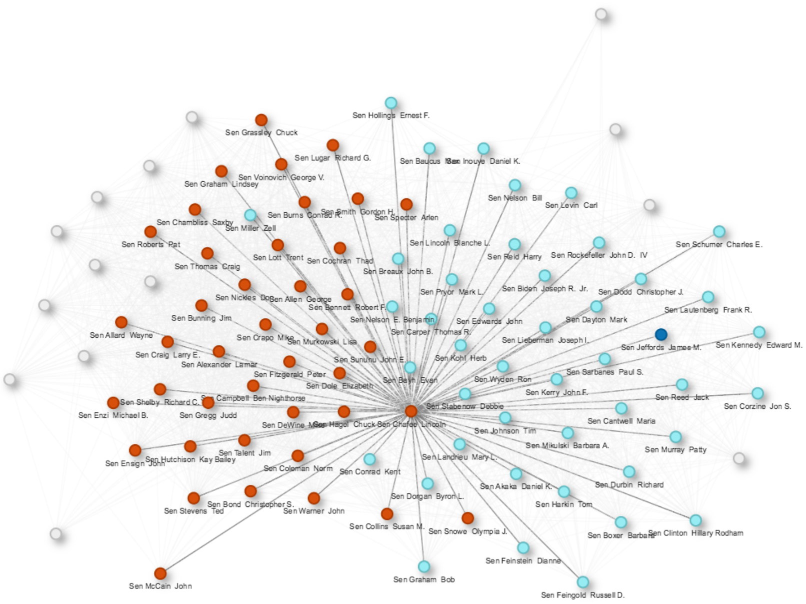


FIGURE A.3: The backbone of the senator-senator projection with senator Lincoln Chafee and his neighbourhood highlighted. Light blue nodes represent democrats, red nodes represent republican members of the senate. The dark blue node represents an independent member of the U.S. Senate.

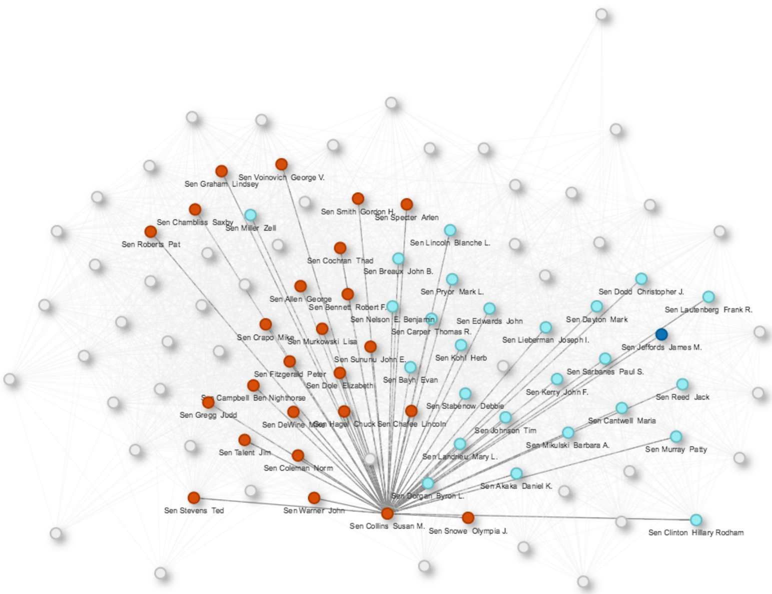


FIGURE A.4: The backbone of the senator-senator projection with senator Susan Collins and his neighbourhood highlighted. Light blue nodes represent democrats, red nodes represent republican members of the senate. The dark blue node represents an independent member of the U.S. Senate.

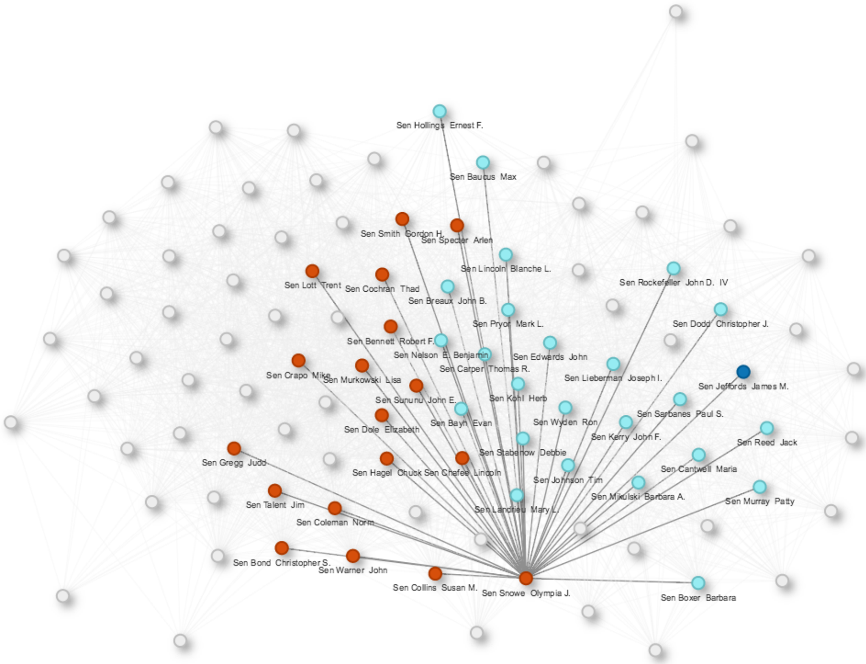


FIGURE A.5: The backbone of the senator-senator projection with senator Olympia Snowe and his neighbourhood highlighted. Light blue nodes represent democrats, red nodes represent republican members of the senate. The dark blue node represents an independent member of the U.S. Senate.

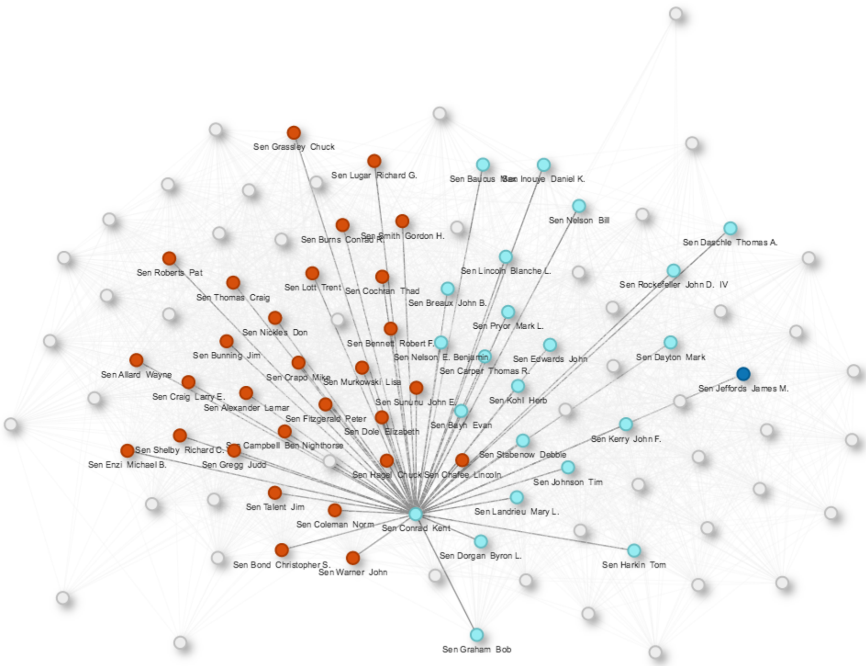


FIGURE A.6: The backbone of the senator-senator projection with senator Kent Conrad and his neighbourhood highlighted. Light blue nodes represent democrats, red nodes represent republican members of the senate. The dark blue node represents an independent member of the U.S. Senate.

Appendix B

Tables

B.1 Communities in the 108th U.S. Senate

Node Id	Senator	Party	[12]	[85]	[101]	[12]	[85]	[101]
1	Akaka Daniel K.	democratic	2	1	1	1	1	1
5	Baucus Max	democratic	1	2	1	1	1	1
6	Bayh Evan	democratic	1	2	1	1	1	1
8	Biden Joseph R. Jr.	democratic	2	1	1	1	1	1
9	Bingaman Jeff	democratic	2	1	1	1	1	1
11	Boxer Barbara	democratic	2	1	1	1	1	1
12	Breaux John B.	democratic	1	2	1	1	1	1
16	Byrd Robert C.	democratic	2	1	1	1	1	1
18	Cantwell Maria	democratic	2	1	1	1	1	1
19	Carper Thomas R.	democratic	2	1	1	1	1	1
22	Clinton Hillary Rodham	democratic	2	1	1	1	1	1
26	Conrad Kent	democratic	1	2	1	2	2	1
28	Corzine Jon S.	democratic	2	1	1	1	1	1
31	Daschle Thomas A.	democratic	2	1	1	1	1	1
32	Dayton Mark	democratic	2	1	1	1	1	1
34	Dodd Christopher J.	democratic	2	1	1	1	1	1
37	Dorgan Byron L.	democratic	2	1	1	1	1	1
38	Durbin Richard	democratic	2	1	1	1	1	1
39	Edwards John	democratic	2	1	1	1	1	1
42	Feingold Russell D.	democratic	2	1	1	1	1	1
43	Feinstein Dianne	democratic	2	1	1	1	1	1
46	Graham Bob	democratic	2	1	1	1	1	1
51	Harkin Tom	democratic	2	1	1	1	1	1
53	Hollings Ernest F.	democratic	1	2	1	1	1	1
56	Inouye Daniel K.	democratic	2	1	1	1	1	1
58	Johnson Tim	democratic	2	1	1	1	1	1

59	Kennedy Edward M.	democratic	2	1	1	1	1	1
60	Kerry John F.	democratic	2	1	1	1	1	1
61	Kohl Herb	democratic	2	1	1	1	1	1
63	Landrieu Mary L.	democratic	2	1	1	1	1	1
64	Lautenberg Frank R.	democratic	2	1	1	1	1	1
65	Leahy Patrick J.	democratic	2	1	1	1	1	1
66	Levin Carl	democratic	2	1	1	1	1	1
67	Lieberman Joseph I.	democratic	2	1	1	1	1	1
68	Lincoln Blanche L.	democratic	2	1	1	1	1	1
73	Mikulski Barbara A.	democratic	2	1	1	1	1	1
74	Miller Zell	democratic	1	2	2	2	2	2
76	Murray Patty	democratic	2	1	1	1	1	1
77	Nelson Bill	democratic	2	1	1	1	1	1
78	Nelson E. Benjamin	democratic	1	2	2	1	1	1
80	Pryor Mark L.	democratic	2	1	1	1	1	1
81	Reed Jack	democratic	2	1	1	1	1	1
82	Reid Harry	democratic	2	1	1	1	1	1
84	Rockefeller John D. IV	democratic	2	1	1	1	1	1
86	Sarbanes Paul S.	democratic	2	1	1	1	1	1
87	Schumer Charles E.	democratic	2	1	1	1	1	1
93	Stabenow Debbie	democratic	2	1	1	1	1	1
100	Wyden Ron	democratic	2	1	1	1	1	1
57	Jeffords James M.	independent	2	1	1	1	1	1
2	Alexander Lamar	republican	1	2	2	2	2	2
3	Allard Wayne	republican	1	2	2	2	2	2
4	Allen George	republican	1	2	2	2	2	2
7	Bennett Robert F.	republican	1	2	2	2	2	1
10	Bond Christopher S.	republican	1	2	2	2	2	2
13	Brownback Sam	republican	1	2	2	2	2	2
14	Bunning Jim	republican	1	2	2	2	2	2
15	Burns Conrad R.	republican	1	2	2	2	2	2
17	Campbell Ben Nighthorse	republican	1	2	2	2	2	2
20	Chafee Lincoln	republican	2	1	1	1	1	1
21	Chambliss Saxby	republican	1	2	2	2	2	2
23	Cochran Thad	republican	1	2	2	2	2	2
24	Coleman Norm	republican	1	2	2	2	2	2
25	Collins Susan M.	republican	2	1	1	1	1	1
27	Cornyn John	republican	1	2	2	2	2	2
29	Craig Larry E.	republican	1	2	2	2	2	2
30	Crapo Mike	republican	1	2	2	2	2	2
33	DeWine Mike	republican	1	2	2	2	2	2
35	Dole Elizabeth	republican	1	2	2	2	2	1
36	Domenici Pete V.	republican	1	2	2	2	2	2
40	Ensign John	republican	1	2	2	2	2	2
41	Enzi Michael B.	republican	1	2	2	2	2	2
44	Fitzgerald Peter	republican	1	2	2	2	2	2
45	Frist William H.	republican	1	2	2	2	2	2
47	Graham Lindsey	republican	1	2	2	2	2	2

48	Grassley Chuck	republican	1	2	2	2	2	2
49	Gregg Judd	republican	1	2	2	2	2	2
50	Hagel Chuck	republican	1	2	2	2	2	2
52	Hatch Orrin G.	republican	1	2	2	2	2	2
54	Hutchison Kay Bailey	republican	1	2	2	2	2	2
55	Inhofe James M.	republican	1	2	2	2	2	2
62	Kyl Jon	republican	1	2	2	2	2	2
69	Lott Trent	republican	1	2	2	2	2	2
70	Lugar Richard G.	republican	1	2	2	2	2	2
71	McCain John	republican	1	2	2	2	2	2
72	McConnell Mitch	republican	1	2	2	2	2	2
75	Murkowski Lisa	republican	1	2	2	2	2	2
79	Nickles Don	republican	1	2	2	2	2	2
83	Roberts Pat	republican	1	2	2	2	2	2
85	Santorum Rick	republican	1	2	2	2	2	2
88	Sessions Jeff	republican	1	2	2	2	2	2
89	Shelby Richard C.	republican	1	2	2	2	2	2
90	Smith Gordon H.	republican	1	2	2	2	2	2
91	Snowe Olympia J.	republican	2	1	1	1	1	1
92	Specter Arlen	republican	1	2	1	2	2	1
94	Stevens Ted	republican	1	2	2	2	2	2
95	Sununu John E.	republican	1	2	2	2	2	1
96	Talent Jim	republican	1	2	2	2	2	2
97	Thomas Craig	republican	1	2	2	2	2	2
98	Voinovich George V.	republican	1	2	2	2	2	2
99	Warner John	republican	1	2	2	2	2	2

Table B.1: The list of senators of the 108$^{\text{th}}$ U.S. Senate and their associated community membership for the three different algorithms in the weighted projection (first three columns) and backbone (last three columns). The column headers give the references to the corresponding community detection algorithms.

B.2 Communities in the MovieLens tag genome network (100 most popular tags)

Node Id	Tag	[12]	[85]	[101]	[12]	[85]	[101]	[12]	[85]	[101]
4	surreal	1	1	1	2	2	2	1	1	3
10	romance	1	1	1	1	1	1	1	4	3
14	thought-provoking	2	2	1	2	2	2	1	5	3
16	quirky	1	1	1	1	1	2	1	1	3
19	visually appealing	1	2	1	2	2	2	1	1	3
20	stylized	1	2	1	2	3	2	1	3	3
24	social commentary	1	2	1	4	2	2	1	5	3
25	drugs	2	2	1	4	3	2	1	3	3

29	music	2	2	1	1	1	1	1	4	6
30	nonlinear	2	2	1	2	2	2	1	3	3
35	cult film	2	2	1	3	3	2	1	1	3
42	great soundtrack	2	2	1	4	2	2	1	5	3
50	bittersweet	2	1	1	4	2	2	1	5	3
55	dreamlike	2	2	1	2	2	2	1	1	3
56	multiple storylines	2	2	1	2	2	2	1	3	3
58	mental illness	2	2	1	4	2	2	1	5	3
64	religion	2	2	1	4	2	2	1	5	3
65	coming of age	1	1	2	4	1	2	1	5	7
68	philosophy	2	2	1	2	2	2	1	1	3
80	mindfuck	2	2	1	2	3	2	1	1	3
90	slow	2	2	1	2	2	2	1	3	3
91	beautiful	2	1	1	4	2	2	1	5	3
93	depressing	2	2	1	4	2	2	1	5	3
94	documentary	2	2	1	4	2	2	1	5	6
98	philosophical	2	2	1	2	2	2	1	1	3
32	boring	1	1	2	1	1	1	2	2	8
1	sci-fi	1	1	2	1	1	1	3	1	5
3	action	1	1	1	1	1	1	3	4	5
12	fantasy	1	1	1	1	1	1	3	1	5
15	time travel	1	1	2	1	1	1	3	1	5
17	dystopia	1	1	1	1	1	2	3	1	5
23	animation	1	1	2	1	1	1	3	4	2
27	adventure	1	1	1	1	1	1	3	4	2
31	predictable	1	1	1	1	1	1	3	4	1
36	post-apocalyptic	1	1	2	1	1	2	3	1	5
37	space	1	1	2	1	1	1	3	1	5
38	aliens	1	1	2	1	1	1	3	1	5
39	alternate reality	1	1	2	1	1	2	3	1	5
48	superhero	1	1	2	1	1	1	3	4	5
53	comic book	1	1	2	1	1	1	3	1	5
57	pixar	1	1	2	1	1	1	3	4	2
62	musical	2	2	1	1	1	1	3	4	2
73	overrated	1	1	2	1	1	1	3	4	1
74	magic	1	1	2	1	1	1	3	4	2
81	martial arts	1	1	1	1	1	1	3	4	5
86	robots	1	1	2	1	1	1	3	1	5
87	family	1	1	2	1	1	1	3	4	2
89	remake	1	1	1	1	1	1	3	4	1
97	fairy tale	1	1	2	1	1	1	3	4	2
99	anime	1	1	2	1	1	2	3	1	5
100	disney	1	1	2	1	1	1	3	4	2
5	twist ending	2	2	1	3	3	2	4	3	3
7	classic	2	2	1	4	2	2	4	5	3
8	atmospheric	2	2	1	3	3	2	4	3	3
11	psychology	2	2	1	3	3	2	4	3	3
18	violence	1	1	1	3	3	2	4	3	3

21	dark	2	2	1	3	3	2	4	3	3
22	disturbing	2	2	1	3	3	2	4	3	3
33	thriller	2	2	1	3	3	2	4	3	3
40	horror	1	1	2	3	3	2	4	3	3
41	nudity (topless)	1	2	1	1	1	2	4	3	3
47	zombies	1	1	2	3	1	2	4	1	3
51	revenge	1	1	1	3	3	2	4	3	3
52	cinematography	1	1	1	4	2	2	4	5	3
54	violent	1	1	1	3	3	2	4	3	3
59	suspense	1	1	1	3	3	2	4	3	3
60	serial killer	2	2	1	3	3	2	4	3	3
61	crime	2	2	1	3	3	2	4	3	3
63	psychological	2	2	1	3	3	2	4	3	3
70	tense	2	2	1	3	3	2	4	3	3
75	black and white	2	2	1	4	2	2	4	5	3
79	mystery	2	2	1	3	3	2	4	3	3
85	organized crime	2	2	1	3	3	2	4	3	3
92	mafia	2	2	1	3	3	2	4	3	3
2	comedy	2	2	1	1	1	1	5	4	4
6	funny	1	1	1	1	1	1	5	4	4
9	dark comedy	2	2	1	1	1	2	5	3	3
26	black comedy	2	2	1	1	1	2	5	3	3
28	satire	2	2	1	1	1	1	5	4	4
66	stupid	1	1	2	1	1	1	5	4	4
69	parody	1	1	2	1	1	1	5	4	4
72	high school	1	1	2	1	1	1	5	5	3
77	witty	2	2	1	1	1	2	5	5	3
78	hilarious	1	1	1	1	1	1	5	4	4
83	humorous	1	1	1	1	1	1	5	4	4
96	coen brothers	2	2	1	1	2	2	5	3	3
13	based on a book	1	1	1	4	2	2	6	5	7
34	true story	2	2	1	4	2	2	6	5	7
43	drama	1	1	1	4	2	2	6	5	7
44	politics	2	2	1	4	2	2	6	5	7
45	war	2	2	1	4	2	2	6	5	7
46	world war ii	2	2	1	4	2	2	6	5	7
49	based on a true story	2	2	1	4	2	2	6	5	7
67	friendship	1	1	1	4	1	2	6	5	7
71	inspirational	1	1	2	4	1	2	6	5	7
76	british	2	2	1	4	1	2	6	5	7
82	imdb top 250	1	2	1	4	2	2	6	5	3
84	history	2	2	1	4	2	2	6	5	7
88	oscar (best picture)	2	2	1	4	2	2	6	5	7
95	great acting	1	1	1	4	2	2	6	5	3

TABLE B.2: The list of the 100 most popular tags in the MovieLens network and their associated community membership for the three different algorithms in the binary projection (first three columns), the weighted projection (columns four, five and six) and the backbone (last three columns).The column headers give the references to the corresponding community detection algorithms.

B.3 Clustering coefficients of users in the MovieLens 10M network

$icc_{i,0}$	$icc_{i,1}$	$icc_{i,2}$	$icc_{i,3}$	$icc_{i,0}$	$icc_{i,1}$	$icc_{i,2}$	$icc_{i,3}$
0.0549	0.1513	0.2043	0.1227	0.0128	0.0291	0.0317	0.0082
0.0869	0.1884	0.2383	0.1512	0.0243	0.0449	0.0420	0.0111
0.0000	0.0138	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0110	0.0000	0.0000	0.0567	0.1113	0.1240	0.0816
0.0383	0.0763	0.0963	0.0456	0.0237	0.0505	0.0659	0.0454
0.0961	0.1682	0.2133	0.1225	0.0657	0.1301	0.1892	0.1193
0.0400	0.0785	0.0820	0.0217	0.0510	0.1019	0.1231	0.0722
0.0658	0.1303	0.1481	0.0785	0.0000	0.0000	0.0000	0.0000
0.0231	0.0664	0.1049	0.0158	0.0018	0.0000	0.0382	0.0000
0.0332	0.0660	0.0841	0.0466	0.0912	0.1710	0.1936	0.1112
0.0212	0.0437	0.0519	0.0258	0.0190	0.0339	0.0593	0.0342
0.0997	0.1885	0.2366	0.1413	0.0930	0.1730	0.2067	0.1175
0.0439	0.0825	0.1004	0.0549	0.0407	0.0831	0.1205	0.0719
0.0379	0.0737	0.0876	0.0497	0.0276	0.0646	0.0760	0.0448
0.0903	0.1711	0.2125	0.1303	0.0209	0.0343	0.0644	0.0377
0.0084	0.0219	0.0481	0.0400	0.0220	0.0526	0.1018	0.0571
0.1160	0.2183	0.2597	0.1614	0.1244	0.2369	0.2873	0.1867
0.0677	0.1015	0.0609	0.0107	0.0461	0.0920	0.1245	0.0711
0.1044	0.1942	0.2445	0.1472	0.0229	0.0478	0.0577	0.0372
0.0658	0.1303	0.1481	0.0785	0.0205	0.0404	0.0627	0.0342
0.0658	0.1303	0.1481	0.0785	0.0217	0.0486	0.0938	0.0537
0.0439	0.0825	0.1004	0.0549	0.0171	0.0315	0.0466	0.0233
0.0711	0.1275	0.1735	0.0948	0.0143	0.0302	0.0438	0.0340
0.0669	0.1272	0.1541	0.0852	0.0522	0.1090	0.1375	0.0796
0.0212	0.0436	0.0519	0.0258	0.0606	0.1300	0.1490	0.0853
0.0328	0.0660	0.0932	0.0526	0.0000	0.0000	0.0000	0.0000
0.0239	0.0493	0.0537	0.0208	0.0427	0.0897	0.1150	0.0701
0.1310	0.2369	0.2781	0.1630	0.0478	0.0887	0.1180	0.0688
0.0000	0.0000	0.0000	0.0000	0.0616	0.1219	0.1512	0.0851
0.0740	0.1451	0.1872	0.1192	0.0638	0.1235	0.1640	0.0994
0.0000	0.0000	0.0000	0.0000	0.0214	0.0423	0.0573	0.0186
0.0289	0.0590	0.0480	0.0201	0.0394	0.0826	0.1042	0.0590
0.1091	0.2057	0.2630	0.1638	0.0288	0.0662	0.0970	0.0624
0.0394	0.0725	0.0932	0.0521	0.0204	0.0369	0.0574	0.0292
0.0357	0.0820	0.1154	0.0695	0.0317	0.0611	0.0816	0.0530

0.0361	0.0645	0.0529	0.0207	0.1242	0.2375	0.2813	0.1824
0.0568	0.1167	0.1466	0.0782	0.0329	0.0732	0.0889	0.0509
0.0606	0.1241	0.1391	0.0753	0.0130	0.0289	0.0445	0.0115
0.0565	0.1047	0.1473	0.0828	0.0278	0.0650	0.0764	0.0454
0.0227	0.0466	0.0646	0.0178	0.0864	0.1846	0.1986	0.1315
0.0768	0.1511	0.1804	0.0987	0.1015	0.1992	0.2515	0.1543
0.0713	0.1362	0.1565	0.0833	0.0828	0.1660	0.2063	0.1401
0.0207	0.0359	0.0575	0.0300	0.0223	0.0532	0.0695	0.0251
0.0000	0.0000	0.0000	0.0000	0.1101	0.2117	0.2406	0.1505
0.0000	0.0000	0.0000	0.0000	0.0616	0.1209	0.1808	0.1235
0.0917	0.1720	0.1957	0.1017	0.0288	0.0515	0.0790	0.0427
0.0689	0.1341	0.1600	0.0849	0.0635	0.1346	0.1571	0.1010
0.0409	0.0772	0.0954	0.0540	0.1233	0.2281	0.2706	0.1821
0.0658	0.1303	0.1481	0.0785	0.0544	0.1165	0.1291	0.0673
0.0697	0.1261	0.1235	0.0506	0.0278	0.0606	0.0769	0.0408
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0297	0.0596	0.0757	0.0391	0.0000	0.0020	0.0000	0.0000
0.0069	0.0180	0.0359	0.0000	0.0216	0.0514	0.0633	0.0326
0.0731	0.1352	0.1629	0.0899	0.0168	0.0332	0.0458	0.0265
0.0000	0.0000	0.0000	0.0000	0.1017	0.1980	0.2291	0.1458
0.0487	0.0941	0.1203	0.0662	0.0012	0.0070	0.0091	0.0000
0.0074	0.0192	0.0317	0.0000	0.0152	0.0287	0.0437	0.0204
0.0264	0.0547	0.0625	0.0538	0.0094	0.0292	0.0424	0.0337
0.0096	0.0161	0.0043	0.0000	0.0163	0.0369	0.0499	0.0286
0.0517	0.1033	0.1207	0.0676	0.0000	0.0000	0.0000	0.0000
0.0633	0.1390	0.1699	0.0896	0.0345	0.0624	0.0919	0.0519
0.0894	0.1517	0.2086	0.1276	0.0299	0.0607	0.0841	0.0510
0.0461	0.0920	0.1245	0.0711	0.0669	0.1343	0.1722	0.1035
0.0286	0.0572	0.0661	0.0355	0.0000	0.0059	0.0000	0.0000
0.1048	0.2030	0.2560	0.1486	0.0365	0.0749	0.0896	0.0480
0.0083	0.0169	0.0309	0.0216	0.0424	0.0851	0.1048	0.0642
0.0718	0.0638	0.0000	0.0000	0.1170	0.2036	0.2052	0.1215
0.0323	0.0624	0.0719	0.0347	0.0436	0.0844	0.1100	0.0599
0.0570	0.1039	0.1481	0.0799	0.0400	0.0741	0.1011	0.0577
0.0882	0.1565	0.2111	0.1242	0.0616	0.1235	0.1416	0.0744
0.0083	0.0169	0.0309	0.0216	0.0278	0.0649	0.0762	0.0451
0.0374	0.0750	0.0912	0.0491	0.0000	0.0000	0.0000	0.0000
0.0323	0.0602	0.0815	0.0440	0.0316	0.0724	0.1015	0.0535
0.0197	0.0055	0.0000	0.0000	0.0455	0.0868	0.1070	0.0637
0.0970	0.2121	0.2428	0.1248	0.0000	0.0000	0.0000	0.0000
0.0029	0.0000	0.0000	0.0000	0.0079	0.0057	0.0010	0.0000
0.0515	0.1008	0.1347	0.0799	0.0725	0.1396	0.1842	0.1028
0.0030	0.0000	0.0000	0.0000	0.0419	0.0854	0.1231	0.0625
0.0550	0.1025	0.1374	0.0762	0.0192	0.0388	0.0586	0.0347
0.0040	0.0000	0.0000	0.0000	0.0627	0.1236	0.1577	0.0921
0.0276	0.0526	0.0653	0.0316	0.0238	0.0519	0.0727	0.0489
0.0220	0.0420	0.0599	0.0298	0.0158	0.0355	0.0488	0.0153
0.0054	0.0089	0.0000	0.0000	0.0055	0.0110	0.0141	0.0000

0.0178	0.0322	0.0497	0.0256	0.0899	0.1729	0.2152	0.1469
0.0154	0.0331	0.0409	0.0150	0.0000	0.0000	0.0000	0.0000
0.0318	0.0604	0.0749	0.0421	0.0193	0.0487	0.0658	0.0333
0.1123	0.2073	0.2640	0.1603	0.0000	0.0000	0.0000	0.0000
0.0483	0.0956	0.1117	0.0602	0.0235	0.0489	0.0627	0.0389
0.0000	0.0000	0.0000	0.0000	0.0165	0.0349	0.0447	0.0265
0.0852	0.1526	0.1890	0.1134	0.0312	0.0597	0.0644	0.0244
0.0438	0.0896	0.1033	0.0600	0.0000	0.0000	0.0000	0.0000
0.0976	0.1780	0.2205	0.1437	0.0693	0.1402	0.1709	0.1018
0.0060	0.0081	0.0112	0.0038	0.0000	0.0091	0.0349	0.0000
0.0798	0.1515	0.1650	0.0891	0.0068	0.0122	0.0196	0.0058
0.1221	0.2192	0.2494	0.1463	0.0196	0.0361	0.0883	0.0595
0.0397	0.0798	0.0956	0.0512	0.0284	0.0653	0.0970	0.0622
0.0833	0.1532	0.1863	0.1010	0.0000	0.0000	0.0000	0.0000
0.0503	0.0949	0.1083	0.0562	0.0155	0.0292	0.0490	0.0233
0.0784	0.1480	0.1731	0.1006	0.0995	0.1867	0.2567	0.1666
0.0597	0.1058	0.1477	0.0817	0.0000	0.0000	0.0534	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0482	0.0900	0.0988	0.0511
0.0636	0.1332	0.1670	0.0906	0.0259	0.0553	0.0782	0.0420
0.0189	0.0234	0.0152	0.0040	0.0076	0.0172	0.0238	0.0098
0.0501	0.1019	0.1327	0.0766	0.0157	0.0323	0.0539	0.0457
0.0205	0.0391	0.0529	0.0259	0.1216	0.2148	0.2375	0.1669
0.0720	0.1314	0.1707	0.0991	0.0191	0.0359	0.0428	0.0247
0.0997	0.1803	0.1892	0.1162	0.1264	0.2336	0.2818	0.1729
0.0188	0.0348	0.0477	0.0299	0.0603	0.1263	0.1505	0.0880
0.0341	0.0714	0.0798	0.0485	0.0334	0.0655	0.0807	0.0588
0.0000	0.0000	0.0000	0.0000	0.0527	0.1027	0.1296	0.0803
0.0156	0.0238	0.0342	0.0188	0.0470	0.0534	0.0746	0.1250
0.0772	0.1462	0.1714	0.0949	0.0393	0.0899	0.1377	0.0826
0.0092	0.0007	0.0000	0.0000	0.0220	0.0465	0.0607	0.0390
0.0182	0.0330	0.0508	0.0262	0.0454	0.1000	0.1362	0.0920
0.0544	0.1041	0.1208	0.0604	0.1098	0.1783	0.1884	0.1143
0.0772	0.1525	0.1823	0.1094	0.0634	0.1311	0.1565	0.1014
0.0655	0.1331	0.1692	0.1024	0.0000	0.0000	0.0000	0.0000
0.0267	0.0490	0.0724	0.0347	0.0292	0.0594	0.0849	0.0539
0.0329	0.0661	0.0937	0.0528	0.0284	0.0653	0.0970	0.0622
0.0683	0.1324	0.1624	0.0974	0.0378	0.0742	0.1144	0.0815
0.0274	0.0559	0.0830	0.0430	0.0193	0.0430	0.0554	0.0368
0.0000	0.0063	0.0000	0.0000	0.0382	0.0754	0.0962	0.0571
0.0498	0.0956	0.1024	0.0560	0.0325	0.0665	0.0845	0.0526
0.0350	0.0715	0.0862	0.0424	0.0635	0.1091	0.1529	0.0960
0.0537	0.1051	0.1348	0.0827	0.0850	0.1606	0.1956	0.1193
0.0838	0.1607	0.2024	0.1171	0.0095	0.0208	0.0324	0.0336
0.0052	0.0091	0.0025	0.0000	0.0531	0.0998	0.1320	0.0755
0.1118	0.2065	0.2421	0.1439	0.0000	0.0000	0.0000	0.0000
0.0227	0.0478	0.0696	0.0354	0.0545	0.1064	0.1472	0.0601
0.0405	0.0774	0.1033	0.0565	0.0099	0.0154	0.0191	0.0000

0.0736	0.1444	0.1844	0.1108	0.0034	0.0110	0.0000	0.0000
0.0309	0.0544	0.0801	0.0428	0.0580	0.1182	0.1461	0.0898
0.1062	0.1966	0.2563	0.1561	0.0441	0.0943	0.1229	0.0692
0.0251	0.0368	0.0525	0.0000	0.0203	0.0436	0.0597	0.0397
0.0182	0.0330	0.0508	0.0262	0.0321	0.0670	0.0998	0.0628
0.1210	0.2318	0.2711	0.1805	0.0359	0.0873	0.1088	0.0614
0.0769	0.1406	0.1867	0.1074	0.0850	0.1673	0.2036	0.1247
0.0217	0.0388	0.0569	0.0285	0.0093	0.0191	0.0000	0.0000
0.0114	0.0213	0.0324	0.0170	0.0273	0.0562	0.0771	0.0439
0.0828	0.1526	0.1923	0.1087	0.0138	0.0348	0.0489	0.0229
0.0000	0.0000	0.0000	0.0000	0.0108	0.0539	0.0814	0.0000
0.0063	0.0084	0.0117	0.0040	0.1036	0.1710	0.1827	0.1075
0.1153	0.2218	0.2644	0.1745	0.0302	0.0615	0.0744	0.0453
0.0000	0.0000	0.0000	0.0000	0.1340	0.2385	0.2535	0.1517
0.0130	0.0290	0.0531	0.0288	0.0561	0.1152	0.1745	0.1070
0.0970	0.1766	0.2226	0.1232	0.0585	0.0857	0.0527	0.0130
0.0000	0.0000	0.0000	0.0000	0.0128	0.0456	0.0000	0.0000
0.0094	0.0147	0.0314	0.0180	0.0237	0.0512	0.0707	0.0367
0.0900	0.1633	0.2113	0.1290	0.0123	0.0182	0.0207	0.0000
0.0149	0.0279	0.0267	0.0035	0.0292	0.0594	0.0849	0.0539
0.0000	0.0000	0.0000	0.0000	0.0000	0.0144	0.0000	0.0000
0.0399	0.0816	0.1005	0.0547	0.0359	0.0781	0.1091	0.0641
0.0437	0.0889	0.1091	0.0586	0.0280	0.0653	0.0745	0.0283
0.0000	0.0000	0.0000	0.0000	0.0178	0.0305	0.0483	0.0285
0.0127	0.0272	0.0257	0.0159	0.0825	0.1615	0.1865	0.1096
0.0537	0.1014	0.1222	0.0754	0.0000	0.0000	0.0000	0.0000
0.0394	0.0775	0.1012	0.0575	0.0182	0.0368	0.0458	0.0418
0.0400	0.0845	0.1107	0.0611	0.0631	0.1289	0.1653	0.0918
0.0295	0.0619	0.0785	0.0429	0.0201	0.0442	0.0584	0.0416
0.0111	0.0119	0.0380	0.0000	0.0867	0.1770	0.2072	0.1159
0.1013	0.1982	0.2369	0.1334	0.0353	0.0811	0.1120	0.0000
0.0205	0.0400	0.0550	0.0291	0.0263	0.1599	0.1134	0.0000
0.0221	0.0495	0.0596	0.0311	0.0000	0.0000	0.0000	0.0000
0.0627	0.1020	0.0709	0.0181	0.0195	0.0456	0.0609	0.0333
0.0285	0.0489	0.0196	0.0000	0.0154	0.0313	0.0433	0.0328
0.0063	0.0084	0.0117	0.0040	0.0153	0.0271	0.0390	0.0190
0.0060	0.0081	0.0081	0.0034	0.0000	0.0000	0.0000	0.0000
0.0461	0.0867	0.1112	0.0613	0.0142	0.0131	0.0455	0.0000
0.0587	0.1163	0.1309	0.0677	0.0923	0.1719	0.2311	0.1520
0.0360	0.0689	0.0861	0.0461	0.0247	0.0531	0.0752	0.0417
0.0186	0.0383	0.0528	0.0233	0.0245	0.0511	0.0661	0.0411
0.1155	0.2239	0.2464	0.1426	0.0727	0.1468	0.1785	0.1021
0.0135	0.0244	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0167	0.0393	0.0526	0.0304	0.1406	0.2673	0.3188	0.2057
0.0734	0.1657	0.2187	0.1377	0.0264	0.0538	0.0726	0.0452
0.0466	0.0979	0.1265	0.0741	0.0000	0.0000	0.0000	0.0000
0.0504	0.0923	0.1201	0.0666	0.0348	0.0723	0.0864	0.0504
0.1186	0.2149	0.2674	0.1715	0.0405	0.0834	0.0999	0.0609

0.0088	0.0163	0.0263	0.0245	0.0322	0.0664	0.0851	0.0507
0.0474	0.0896	0.1169	0.0729	0.0531	0.1203	0.1567	0.0852
0.0209	0.0383	0.0432	0.0179	0.0000	0.0000	0.0000	0.0000
0.0131	0.0182	0.0000	0.0000	0.0586	0.1157	0.1300	0.0703
0.0129	0.0179	0.0000	0.0000	0.0205	0.0428	0.0520	0.0267
0.0000	0.0000	0.0000	0.0000	0.0179	0.0353	0.0582	0.0231
0.0111	0.0235	0.0313	0.0189	0.0572	0.1131	0.1473	0.0906
0.0636	0.1269	0.1484	0.0850	0.0231	0.0472	0.0598	0.0343
0.0275	0.0595	0.0801	0.0456	0.0696	0.1389	0.1819	0.1106
0.1014	0.1982	0.2369	0.1334	0.0282	0.0593	0.0731	0.0336
0.0189	0.0438	0.0559	0.0317	0.0049	0.0050	0.0054	0.0000
0.0571	0.1117	0.1384	0.0806	0.0143	0.0369	0.0740	0.0579
0.0281	0.0587	0.0708	0.0393	0.0352	0.0791	0.1053	0.0628
0.0000	0.0000	0.0000	0.0000	0.0993	0.2100	0.2589	0.1520
0.0064	0.0075	0.0149	0.0000	0.0058	0.0265	0.0655	0.0322
0.0525	0.0966	0.1041	0.0631	0.0229	0.0460	0.0585	0.0343
0.0263	0.0524	0.0727	0.0409	0.0020	0.0100	0.0369	0.0348
0.0184	0.0384	0.0413	0.0202	0.0247	0.0489	0.0600	0.0363
0.0297	0.0566	0.0621	0.0305	0.0302	0.0593	0.0786	0.0537
0.1217	0.2235	0.2669	0.1683	0.0072	0.0697	0.1026	0.0683
0.0899	0.1816	0.2003	0.1146	0.0738	0.1319	0.1973	0.1304
0.0306	0.0625	0.0750	0.0455	0.0369	0.0762	0.1055	0.0591
0.0306	0.0625	0.0750	0.0455	0.0353	0.0729	0.1017	0.0575
0.0079	0.0121	0.0000	0.0000	0.0204	0.0436	0.0568	0.0380
0.0000	0.0000	0.0000	0.0000	0.0089	0.0239	0.0327	0.0115
0.0405	0.0836	0.1149	0.0580	0.0110	0.0138	0.0000	0.0000
0.0852	0.1628	0.1987	0.1188	0.0407	0.0794	0.1007	0.0655
0.0134	0.0293	0.0554	0.0190	0.0712	0.1459	0.1760	0.1111
0.0424	0.0764	0.0754	0.0287	0.0800	0.1537	0.1744	0.1051
0.1093	0.2132	0.2624	0.1533	0.0341	0.0667	0.0811	0.0550
0.0570	0.1103	0.1343	0.0801	0.0325	0.0670	0.0859	0.0512
0.0092	0.0203	0.0299	0.0178	0.0229	0.0460	0.0585	0.0343
0.0453	0.0973	0.0884	0.0360	0.0000	0.0000	0.0000	0.0000
0.0394	0.0800	0.0895	0.0507	0.0302	0.0593	0.0786	0.0537
0.0825	0.1437	0.1828	0.1182	0.0985	0.1783	0.2205	0.1373
0.0056	0.0083	0.0061	0.0012	0.0802	0.1594	0.2013	0.1105
0.0570	0.1103	0.1343	0.0801	0.0000	0.0000	0.0000	0.0000
0.0657	0.1185	0.1473	0.0865	0.0615	0.1162	0.1461	0.0896
0.0178	0.0000	0.0000	0.0000	0.1136	0.2255	0.2610	0.1638
0.0092	0.0203	0.0299	0.0178	0.0586	0.1089	0.1524	0.0893
0.0161	0.0284	0.0427	0.0267	0.0400	0.0798	0.0982	0.0537
0.0482	0.0968	0.1187	0.0704	0.0756	0.1438	0.1677	0.0966
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0421	0.0819	0.0914	0.0500	0.0065	0.0000	0.0000	0.0000
0.0071	0.0402	0.0773	0.0419	0.0032	0.0089	0.0121	0.0000
0.0392	0.0769	0.0884	0.0422	0.0119	0.0474	0.0000	0.0000
0.0186	0.0336	0.0459	0.0221	0.1461	0.2307	0.2295	0.1271
0.0439	0.0876	0.1066	0.0569	0.1008	0.2053	0.2466	0.1562

0.0567	0.1104	0.1229	0.0675	0.0099	0.0240	0.0332	0.0214
0.0000	0.0287	0.0761	0.1071	0.0486	0.1021	0.1201	0.0656
0.0579	0.0996	0.1295	0.0661	0.0000	0.0000	0.0000	0.0000
0.0896	0.1336	0.2075	0.1288	0.0784	0.1516	0.1849	0.1040
0.0124	0.0267	0.0370	0.0233	0.0572	0.1131	0.1473	0.0906
0.0449	0.0960	0.1162	0.0658	0.0683	0.1350	0.1587	0.0964
0.0470	0.0792	0.1169	0.0668	0.0184	0.0384	0.0413	0.0202
0.0000	0.0000	0.0000	0.0000	0.0764	0.1537	0.1788	0.1062
0.0184	0.0366	0.0411	0.0158	0.0000	0.0000	0.0000	0.0000
0.0203	0.0384	0.0443	0.0220	0.1041	0.1982	0.2242	0.1399
0.0994	0.1892	0.2246	0.1334	0.0000	0.0000	0.0000	0.0000
0.0184	0.0366	0.0411	0.0158	0.0966	0.1931	0.2303	0.1380
0.0184	0.0367	0.0411	0.0158	0.0649	0.1264	0.1488	0.0906
0.0730	0.1411	0.1625	0.0869	0.0128	0.0384	0.0694	0.0222
0.0716	0.1373	0.1733	0.1012	0.0170	0.0371	0.0498	0.0282
0.0040	0.0000	0.0000	0.0000	0.0461	0.0965	0.1187	0.0668
0.0137	0.0207	0.0277	0.0050	0.0000	0.0000	0.0000	0.0000
0.0981	0.1852	0.2273	0.1318	0.0000	0.0000	0.0000	0.0000
0.0064	0.0123	0.0060	0.0000	0.0681	0.1363	0.1639	0.0962
0.0010	0.0000	0.0000	0.0000	0.0301	0.0632	0.0785	0.0441
0.0672	0.1362	0.1753	0.0992	0.1025	0.1952	0.2208	0.1378
0.0803	0.1539	0.1772	0.1040	0.1449	0.2678	0.2929	0.1797
0.0103	0.0266	0.0227	0.0143	0.0168	0.0295	0.0438	0.0312
0.0020	0.0235	0.0282	0.0194	0.0151	0.0330	0.0588	0.0508
0.0618	0.1176	0.1333	0.0833	0.0000	0.0000	0.0000	0.0000
0.1164	0.2155	0.2284	0.1293	0.0000	0.0000	0.0000	0.0000
0.0034	0.0164	0.0513	0.0000	0.0202	0.0413	0.0511	0.0333
0.0684	0.1337	0.1482	0.0771	0.0133	0.0256	0.0350	0.0281
0.0382	0.0788	0.0848	0.0417	0.0128	0.0438	0.0567	0.0119
0.0382	0.0741	0.0964	0.0536	0.0302	0.0632	0.0936	0.0515
0.0060	0.0082	0.0081	0.0034	0.0000	0.0116	0.0000	0.0000
0.0212	0.0410	0.0436	0.0237	0.0145	0.0261	0.0415	0.0219
0.0375	0.0601	0.0868	0.0530	0.0441	0.0903	0.1333	0.0763
0.0156	0.0276	0.0399	0.0174	0.0208	0.0466	0.0658	0.0354
0.0011	0.0718	0.1157	0.0271	0.0141	0.0430	0.0784	0.0579
0.0345	0.0667	0.0787	0.0466	0.0337	0.0538	0.0000	0.0000
0.0450	0.1027	0.1114	0.0542	0.0225	0.0488	0.0589	0.0333
0.0107	0.0121	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0983	0.1796	0.2847	0.1928	0.0483	0.0988	0.1225	0.0677
0.1160	0.2183	0.2597	0.1614	0.0230	0.0519	0.0688	0.0408
0.0381	0.0740	0.0914	0.0552	0.0171	0.0366	0.0396	0.0224
0.0696	0.1361	0.1617	0.1061	0.0236	0.0456	0.0547	0.0416
0.0130	0.0119	0.0000	0.0000	0.0333	0.0697	0.0773	0.0458
0.0383	0.0748	0.0837	0.0511	0.0355	0.0738	0.0834	0.0515
0.0339	0.0685	0.0759	0.0413	0.0127	0.0276	0.0352	0.0245
0.0113	0.0206	0.0339	0.0214	0.0340	0.0717	0.0795	0.0458
0.0296	0.0636	0.0710	0.0377	0.0000	0.0000	0.0000	0.0000
0.0060	0.0081	0.0081	0.0034	0.0349	0.0736	0.0816	0.0470

0.0135	0.0274	0.0387	0.0219	0.0374	0.0789	0.0874	0.0503
0.0258	0.0518	0.0685	0.0369	0.0358	0.0742	0.0955	0.0554
0.0166	0.0000	0.0000	0.0000	0.0524	0.1054	0.1383	0.0790
0.0000	0.0283	0.0502	0.0824	0.0565	0.1158	0.1537	0.1014
0.0910	0.1654	0.2315	0.1402	0.0315	0.0720	0.0760	0.0359
0.0372	0.0735	0.0922	0.0538	0.0388	0.0827	0.1099	0.0629
0.0604	0.1154	0.1459	0.0831	0.0235	0.0491	0.0645	0.0348
0.0783	0.1383	0.1977	0.1177	0.0306	0.0625	0.0750	0.0455
0.0175	0.0379	0.0450	0.0212	0.0383	0.0811	0.0995	0.0552
0.0215	0.0422	0.0000	0.0000	0.0304	0.0575	0.0812	0.0493
0.0817	0.1615	0.1934	0.1141	0.0368	0.0713	0.0910	0.0499
0.0125	0.0221	0.0249	0.0000	0.0572	0.1131	0.1473	0.0906
0.0544	0.1108	0.1198	0.0638	0.0308	0.0600	0.0780	0.0473
0.0753	0.1480	0.1744	0.1021	0.0802	0.1594	0.2013	0.1105
0.0000	0.0000	0.0000	0.0000	0.0230	0.0519	0.0688	0.0408
0.0352	0.0703	0.0798	0.0422	0.0327	0.0682	0.0692	0.0369
0.0079	0.0204	0.0310	0.0190	0.0180	0.0488	0.0540	0.0157
0.0000	0.0000	0.0000	0.0000	0.0000	0.0138	0.0555	0.0462
0.0390	0.0724	0.0848	0.0416	0.0154	0.0525	0.0415	0.0134
0.0863	0.1606	0.1825	0.1020	0.0209	0.0435	0.0462	0.0380
0.0438	0.0882	0.1395	0.0603	0.0356	0.0732	0.0937	0.0585
0.0324	0.0773	0.1167	0.0599	0.0638	0.1255	0.1551	0.1000
0.0317	0.0598	0.0611	0.0227	0.0696	0.1331	0.1717	0.0996
0.0317	0.0554	0.0935	0.0413	0.0000	0.0000	0.0000	0.0000
0.0153	0.0000	0.0000	0.0000	0.0354	0.0730	0.0859	0.0472
0.0171	0.0335	0.0375	0.0208	0.0032	0.0089	0.0121	0.0000
0.0140	0.0305	0.0288	0.0000	0.0090	0.0208	0.0277	0.0152
0.0223	0.0359	0.0579	0.0253	0.0368	0.0665	0.0587	0.0151
0.0028	0.0042	0.0000	0.0000	0.0552	0.1063	0.1265	0.0774
0.0125	0.0162	0.0233	0.0090	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0254	0.0606	0.0660	0.0320
0.0263	0.0504	0.0603	0.0324	0.0297	0.0682	0.0721	0.0344
0.0625	0.1149	0.1407	0.0740	0.0247	0.0591	0.0642	0.0311
0.0258	0.0603	0.0636	0.0335	0.0247	0.0591	0.0642	0.0311
0.0432	0.0000	0.0000	0.0000	0.0135	0.0199	0.0351	0.0132
0.0741	0.1431	0.1566	0.0908	0.0896	0.1748	0.2129	0.1293
0.0198	0.0383	0.0376	0.0201	0.0490	0.1311	0.1749	0.0833
0.0455	0.0935	0.1086	0.0573	0.0693	0.1267	0.1765	0.1083
0.0000	0.0000	0.0000	0.0000	0.0450	0.0953	0.1193	0.0626
0.0316	0.0589	0.0627	0.0339	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0389	0.0886	0.1235	0.0629
0.0304	0.0621	0.0746	0.0452	0.0383	0.0811	0.0995	0.0552
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0316	0.0618	0.1028	0.0616
0.0114	0.0256	0.0345	0.0231	0.0000	0.0000	0.0000	0.0000
0.0352	0.0784	0.1030	0.0481	0.0568	0.1319	0.1638	0.0915
0.0230	0.0422	0.0541	0.0261	0.0011	0.0024	0.0000	0.0000
0.0577	0.1008	0.1420	0.0871	0.0032	0.0089	0.0121	0.0000

0.0000	0.0000	0.0000	0.0000	0.0115	0.0243	0.0301	0.0185
0.0299	0.0531	0.0672	0.0335	0.0369	0.0713	0.0970	0.0563
0.0555	0.1120	0.1422	0.0866	0.0579	0.1118	0.1640	0.0887
0.0211	0.0385	0.0561	0.0264	0.0894	0.1698	0.1927	0.0864
0.0549	0.1061	0.1245	0.0659	0.0616	0.1202	0.1425	0.0832
0.0364	0.0768	0.0914	0.0486	0.0335	0.0656	0.0905	0.0535
0.0354	0.0721	0.0959	0.0541	0.0101	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0322	0.0691	0.1084	0.0646
0.0799	0.1903	0.2262	0.1614	0.0151	0.0354	0.0481	0.0302
0.0781	0.1458	0.1644	0.0807	0.0000	0.0000	0.0000	0.0000
0.0435	0.0892	0.1001	0.0576	0.0000	0.0030	0.0000	0.0000
0.0285	0.0500	0.0742	0.0415	0.0346	0.0646	0.0862	0.0497
0.0000	0.0000	0.0000	0.0000	0.0134	0.0314	0.0282	0.0102
0.0784	0.1383	0.1768	0.1100	0.0088	0.0000	0.0000	0.0000
0.0294	0.0369	0.0484	0.0000	0.0068	0.0133	0.0196	0.0000
0.0000	0.0000	0.0000	0.0000	0.0069	0.0137	0.0196	0.0000
0.0606	0.1150	0.1478	0.0893	0.0465	0.0955	0.1191	0.0895
0.0722	0.1378	0.1785	0.1009	0.0011	0.0023	0.0056	0.0000
0.0703	0.1287	0.1709	0.1016	0.0176	0.0381	0.0512	0.0291
0.0000	0.0000	0.0000	0.0000	0.0174	0.0378	0.0507	0.0288
0.0297	0.0545	0.0755	0.0431	0.0743	0.1416	0.1523	0.0872
0.0667	0.1275	0.1561	0.0871	0.0000	0.0000	0.0000	0.0000
0.0438	0.0785	0.1207	0.0738	0.0101	0.0011	0.0070	0.0000
0.0716	0.1281	0.1527	0.0927	0.0102	0.0012	0.0070	0.0000
0.0168	0.0359	0.0419	0.0222	0.0477	0.1004	0.1276	0.0748
0.0913	0.1828	0.2447	0.1506	0.0720	0.1388	0.1695	0.1045
0.0000	0.0000	0.0000	0.0000	0.0857	0.1471	0.2199	0.1422
0.0422	0.0725	0.0924	0.0458	0.0139	0.0218	0.0336	0.0288
0.0145	0.0329	0.0419	0.0244	0.0134	0.0308	0.0488	0.0296
0.0138	0.0257	0.0421	0.0226	0.0079	0.0076	0.0000	0.0000
0.0156	0.0316	0.0431	0.0225	0.0152	0.0349	0.0533	0.0408
0.0000	0.0000	0.0000	0.0000	0.1118	0.2186	0.2797	0.1733
0.1193	0.2385	0.2970	0.1821	0.0404	0.0811	0.1011	0.0576
0.0282	0.0558	0.0814	0.0516	0.0000	0.0000	0.0000	0.0000
0.0390	0.0724	0.0931	0.0618	0.0327	0.0668	0.0731	0.0372
0.1054	0.1988	0.2644	0.1566	0.0300	0.0629	0.0835	0.0486
0.0436	0.0838	0.0955	0.0526	0.0199	0.0381	0.0522	0.0303
0.0222	0.0441	0.0580	0.0334	0.0116	0.0117	0.0000	0.0000
0.0258	0.0565	0.0732	0.0381	0.0356	0.0748	0.0984	0.0570
0.0643	0.1244	0.1443	0.0788	0.0292	0.0586	0.0821	0.0460
0.0202	0.0454	0.0607	0.0314	0.0207	0.0371	0.0449	0.0240
0.0148	0.0299	0.0489	0.0323	0.0144	0.0196	0.0213	0.0078
0.0460	0.1009	0.1080	0.0587	0.0152	0.0380	0.0699	0.0616
0.0408	0.0842	0.1013	0.0543	0.0912	0.1712	0.2105	0.1135
0.0549	0.1069	0.1414	0.0799	0.0144	0.0196	0.0213	0.0078
0.0279	0.0496	0.0610	0.0000	0.0000	0.0000	0.0000	0.0000
0.0304	0.0621	0.0746	0.0452	0.0144	0.0196	0.0213	0.0078
0.0387	0.0960	0.1157	0.0548	0.0349	0.0736	0.1079	0.0718

0.0000	0.0000	0.0180	0.0000	0.0138	0.0286	0.0390	0.0264
0.0000	0.0000	0.0000	0.0000	0.0774	0.1559	0.1841	0.0927
0.0000	0.0322	0.0000	0.0000	0.0870	0.1723	0.1986	0.1067
0.0123	0.0144	0.0000	0.0000	0.0870	0.1723	0.1986	0.1067
0.0405	0.0810	0.0954	0.0504	0.0000	0.0000	0.0000	0.0000
0.0480	0.0705	0.0502	0.0000	0.0128	0.0298	0.0429	0.0130
0.0293	0.0573	0.0685	0.0342	0.0000	0.0000	0.0000	0.0000
0.0428	0.0675	0.0563	0.0132	0.0128	0.0296	0.0429	0.0130
0.0000	0.0000	0.0000	0.0000	0.0040	0.0160	0.0384	0.0000
0.0602	0.1077	0.1468	0.0994	0.0193	0.0367	0.0469	0.0216
0.0604	0.1216	0.1516	0.0856	0.0688	0.1237	0.1415	0.0741
0.0173	0.0114	0.0000	0.0000	0.0229	0.0856	0.1669	0.0652
0.1134	0.1891	0.2098	0.1234	0.0164	0.0304	0.0406	0.0223
0.0624	0.1127	0.1374	0.0766	0.0229	0.0557	0.0708	0.0217
0.0671	0.1007	0.1279	0.0594	0.0757	0.1478	0.1683	0.0902
0.1096	0.2040	0.2380	0.1340	0.0213	0.0526	0.0668	0.0205
0.0836	0.1626	0.1933	0.1167	0.0381	0.0717	0.0902	0.0259
0.0976	0.1752	0.2334	0.1433	0.0698	0.1373	0.1601	0.0971
0.0276	0.0537	0.0715	0.0411	0.0197	0.0357	0.0440	0.0192
0.0428	0.0675	0.0563	0.0132	0.0089	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0096	0.0209	0.0000	0.0000
0.0462	0.0822	0.1167	0.0716	0.0155	0.0267	0.0394	0.0237
0.0306	0.0610	0.0769	0.0443	0.0104	0.0154	0.0257	0.0097
0.0000	0.0000	0.0000	0.0000	0.0915	0.1656	0.1855	0.1077
0.0072	0.0105	0.0107	0.0000	0.0866	0.1748	0.1964	0.1039
0.0503	0.0927	0.1077	0.0663	0.0182	0.0458	0.0666	0.0402
0.0000	0.0000	0.0000	0.0000	0.0126	0.0189	0.0336	0.0263
0.0317	0.0611	0.0816	0.0530	0.0719	0.1426	0.1671	0.1111
0.0521	0.1019	0.1168	0.0661	0.0635	0.1268	0.1456	0.0820
0.0176	0.0349	0.0395	0.0188	0.0635	0.1268	0.1456	0.0820
0.0173	0.0392	0.0587	0.0288	0.0969	0.1790	0.2232	0.1336
0.0280	0.0539	0.0670	0.0334	0.0581	0.1244	0.1412	0.0765
0.0112	0.0308	0.0373	0.0045	0.0266	0.0571	0.0680	0.0411
0.1162	0.2248	0.2541	0.1523	0.0182	0.0406	0.0550	0.0317
0.0342	0.0702	0.0926	0.0620	0.0635	0.1268	0.1456	0.0820
0.0064	0.0086	0.0135	0.0000	0.0227	0.0420	0.0561	0.0383
0.0000	0.0000	0.0000	0.0000	0.0885	0.1660	0.1841	0.1034
0.0000	0.0000	0.0000	0.0000	0.0247	0.0529	0.0695	0.0353
0.0000	0.0000	0.0000	0.0000	0.0219	0.0357	0.0572	0.0373
0.0327	0.0644	0.0910	0.0621	0.0227	0.0465	0.0569	0.0334
0.0296	0.0426	0.0594	0.0306	0.0173	0.0337	0.0473	0.0631
0.0482	0.1009	0.1195	0.0667	0.0287	0.0610	0.0590	0.0291
0.0730	0.1410	0.1901	0.1129	0.0311	0.0609	0.0723	0.0480
0.0117	0.0223	0.0344	0.0325	0.0218	0.0423	0.0534	0.0410
0.0657	0.1242	0.1433	0.0873	0.0218	0.0423	0.0534	0.0410
0.0547	0.0935	0.1292	0.0757	0.0348	0.0703	0.0937	0.0529
0.0140	0.0216	0.0328	0.0099	0.0173	0.0370	0.0487	0.0292
0.0716	0.1470	0.1893	0.1078	0.0257	0.0534	0.0677	0.0368

0.0353	0.0683	0.0819	0.0510	0.0280	0.0547	0.0649	0.0330
0.0652	0.1182	0.1533	0.0892	0.0335	0.0474	0.0427	0.0319
0.0493	0.0986	0.1317	0.0817	0.0310	0.0638	0.0793	0.0441
0.0392	0.0726	0.1097	0.0672	0.0300	0.0586	0.0726	0.0405
0.0244	0.0393	0.0474	0.0220	0.0172	0.0331	0.0554	0.0333
0.0291	0.0556	0.0728	0.0372	0.0322	0.0633	0.0802	0.0444
0.0486	0.1187	0.1946	0.1465	0.0329	0.0622	0.0891	0.0488
0.0828	0.1688	0.2074	0.1251	0.0210	0.0424	0.0553	0.0286
0.0398	0.0817	0.0937	0.0568	0.0074	0.0247	0.0111	0.0000
0.0866	0.1679	0.2121	0.1389	0.0149	0.0474	0.0398	0.0000
0.0734	0.1292	0.1635	0.0972	0.0179	0.0297	0.0496	0.0250
0.0384	0.0768	0.0924	0.0549	0.0192	0.0358	0.0545	0.0320
0.0271	0.0561	0.0717	0.0386	0.0138	0.0805	0.1603	0.0585
0.0899	0.1830	0.2249	0.1360	0.0032	0.0050	0.0819	0.0000
0.0737	0.1390	0.1631	0.0968	0.0493	0.0885	0.1088	0.0619
0.0642	0.1033	0.1158	0.0641	0.0636	0.1118	0.1319	0.0713
0.0325	0.0640	0.0863	0.0475	0.0585	0.1195	0.1365	0.0683
0.0309	0.0689	0.0560	0.0168	0.0271	0.0544	0.0682	0.0343
0.0435	0.0837	0.1010	0.0611	0.0586	0.1090	0.1305	0.0675
0.0500	0.0961	0.1117	0.0668	0.0735	0.1382	0.1669	0.0931
0.0435	0.0837	0.1010	0.0611	0.0221	0.0555	0.0625	0.0321
0.0345	0.0700	0.0924	0.0623	0.0238	0.0000	0.0000	0.0000
0.0469	0.0960	0.1023	0.0610	0.0206	0.0443	0.0521	0.0241
0.0154	0.0279	0.0468	0.0276	0.0398	0.0736	0.0759	0.0360
0.0238	0.0518	0.0726	0.0489	0.0501	0.0996	0.1190	0.0652
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0251	0.0531	0.0675	0.0360	0.0933	0.1773	0.2040	0.1094
0.0771	0.1253	0.1319	0.0617	0.0773	0.1476	0.1757	0.1011
0.0276	0.0550	0.0678	0.0359	0.0362	0.0739	0.1099	0.0575
0.0232	0.0532	0.0649	0.0514	0.0000	0.0000	0.0000	0.0000
0.0309	0.0630	0.0758	0.0459	0.0072	0.0149	0.0298	0.0000
0.0068	0.0223	0.0443	0.0269	0.0674	0.1324	0.1656	0.0886
0.0469	0.0922	0.1204	0.0698	0.0065	0.0164	0.0401	0.0000
0.0269	0.0529	0.0753	0.0455	0.0427	0.0876	0.1067	0.0581
0.0445	0.0848	0.1077	0.0599	0.0424	0.0828	0.1019	0.0550
0.0000	0.0006	0.0000	0.0000	0.0000	0.0538	0.0000	0.0000
0.0158	0.0320	0.0458	0.0443	0.0377	0.0688	0.0881	0.0483
0.0617	0.0927	0.1280	0.0695	0.0000	0.0000	0.0000	0.0000
0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.0201	0.0276	0.0459	0.0231	0.0621	0.1236	0.1507	0.0761
0.0019	0.0000	0.0000	0.0000	0.0757	0.1421	0.1739	0.0888
0.0138	0.0241	0.0372	0.0236	0.0160	0.0582	0.0461	0.0191
0.0138	0.0241	0.0369	0.0234	0.0505	0.0968	0.1102	0.0636
0.1288	0.2235	0.2601	0.1732	0.0090	0.0000	0.0000	0.0000
0.0309	0.0628	0.0754	0.0463	0.1310	0.2438	0.2764	0.1719
0.0364	0.0814	0.1102	0.0691	0.0445	0.0980	0.1149	0.0561
0.0172	0.0290	0.0491	0.0281	0.1073	0.2060	0.2138	0.1309
0.0167	0.0333	0.0442	0.0257	0.0335	0.0667	0.0757	0.0385

0.0272	0.0536	0.0604	0.0321	0.0132	0.0302	0.0448	0.0242
0.0135	0.0238	0.0366	0.0233	0.0371	0.0707	0.0801	0.0415
0.0000	0.0324	0.0855	0.0000	0.1112	0.2145	0.2560	0.1508
0.0256	0.0503	0.0584	0.0346	0.0035	0.0057	0.0078	0.0018
0.0090	0.0355	0.0694	0.0223	0.0915	0.1694	0.1967	0.1146
0.0422	0.0639	0.1135	0.0751	0.0798	0.1536	0.1781	0.0995
0.0539	0.1071	0.1384	0.0865	0.1112	0.2145	0.2560	0.1508
0.0348	0.0813	0.1012	0.0661	0.0506	0.0937	0.1179	0.0606
0.1315	0.2537	0.3011	0.1908	0.0356	0.0709	0.0797	0.0500
0.0010	0.0049	0.0000	0.0000	0.0633	0.1261	0.1470	0.0783
0.0877	0.1736	0.1956	0.1160	0.0411	0.0825	0.0882	0.0471
0.0688	0.1532	0.1975	0.1292	0.0545	0.1047	0.1359	0.0670
0.0360	0.0690	0.0897	0.0445	0.0315	0.0617	0.0767	0.0376
0.0256	0.0477	0.0630	0.0398	0.0000	0.0000	0.0000	0.0000

TABLE B.3: The local bipartite clustering coefficients of users who were the first to rate a new movie.

B.4 Clustering coefficients of users in the Digg network

$icc_{i,0}$	$icc_{i,1}$	$icc_{i,2}$	$icc_{i,3}$	$icc_{i,0}$	$icc_{i,1}$	$icc_{i,2}$	$icc_{i,3}$
0.0386	0.0783	0.0946	0.0392	0.0482	0.1062	0.1389	0.0699
0.0926	0.1827	0.1806	0.0860	0.1165	0.2381	0.2607	0.1374
0.0658	0.1504	0.1941	0.1027	0.0685	0.1343	0.1426	0.0665
0.0367	0.0880	0.1119	0.0497	0.0523	0.1058	0.1131	0.0484
0.0331	0.0710	0.0815	0.0321	0.0428	0.0883	0.1066	0.0459
0.0518	0.0795	0.1126	0.0691	0.0559	0.1193	0.1432	0.0666
0.0716	0.1441	0.1480	0.0738	0.0554	0.1121	0.1271	0.0557
0.1260	0.2225	0.2332	0.1295	0.0584	0.1229	0.1425	0.0675
0.0186	0.0294	0.0351	0.0549	0.0271	0.0574	0.0723	0.0303
0.0301	0.0650	0.0780	0.0308	0.1273	0.2650	0.2736	0.1238
0.0728	0.1328	0.1313	0.0569	0.0508	0.0996	0.1216	0.0572
0.0404	0.0757	0.0894	0.0380	0.0861	0.1566	0.1622	0.0802
0.0396	0.0829	0.0928	0.0391	0.0498	0.0978	0.1229	0.0519
0.0337	0.0702	0.0853	0.0353	0.0607	0.1332	0.1894	0.0890
0.0360	0.0717	0.0870	0.0356	0.0333	0.0800	0.1128	0.0550
0.0587	0.1201	0.1351	0.0641	0.1058	0.1908	0.1979	0.0958
0.0861	0.1803	0.2240	0.1240	0.0906	0.1712	0.1782	0.0890
0.0000	0.0000	0.0951	0.0000	0.0532	0.1101	0.1305	0.0649
0.1000	0.1795	0.2031	0.1124	0.0422	0.0869	0.1030	0.0438
0.0288	0.0568	0.0706	0.0262	0.0436	0.0874	0.1037	0.0463
0.0483	0.0953	0.1155	0.0522	0.0268	0.0578	0.0760	0.0323
0.1178	0.2153	0.2178	0.1065	0.0488	0.0925	0.1036	0.0479
0.0997	0.1793	0.1921	0.1042	0.0517	0.1163	0.1473	0.0735
0.0853	0.1595	0.1864	0.1018	0.0686	0.1382	0.1448	0.0727

0.0486	0.1035	0.1262	0.0543	0.0518	0.0996	0.1238	0.0580
0.0417	0.0737	0.0950	0.0405	0.0955	0.1815	0.1790	0.0901
0.0574	0.1256	0.1495	0.0713	0.0959	0.1729	0.2035	0.0940
0.0513	0.0950	0.1137	0.0498	0.0424	0.0911	0.1134	0.0594
0.1238	0.2257	0.2348	0.1253	0.0424	0.0959	0.1194	0.0557
0.0468	0.0996	0.1173	0.0511	0.0599	0.1328	0.1611	0.0849
0.0517	0.1031	0.1153	0.0490	0.0450	0.0874	0.1144	0.0529
0.0619	0.1322	0.1582	0.0907	0.0358	0.0804	0.0979	0.0408
0.0356	0.0770	0.0902	0.0359	0.0547	0.1077	0.1307	0.0618
0.0285	0.0568	0.0769	0.0351	0.0612	0.1250	0.1493	0.0788
0.0269	0.0529	0.0671	0.0242	0.0419	0.0837	0.0993	0.0381
0.0542	0.1196	0.1435	0.0661	0.0535	0.1059	0.1331	0.0655
0.0739	0.1552	0.1826	0.0902	0.0000	0.0000	0.0000	0.0000
0.0624	0.1304	0.1820	0.1006	0.0397	0.0718	0.0952	0.0413
0.0761	0.1514	0.1784	0.0863	0.0514	0.1003	0.1558	0.0817
0.0337	0.0624	0.0810	0.0330	0.0456	0.0878	0.0994	0.0352
0.0345	0.0630	0.0835	0.0349	0.0419	0.1034	0.1418	0.0658
0.0915	0.1735	0.1786	0.0887	0.0609	0.1412	0.1777	0.0905
0.0369	0.0713	0.0830	0.0327	0.0550	0.1092	0.1599	0.0888
0.1173	0.2007	0.2046	0.1022	0.0973	0.1854	0.1894	0.0910
0.0627	0.1204	0.1307	0.0598	0.0603	0.1242	0.1476	0.0684
0.0683	0.1198	0.1284	0.0613	0.1314	0.2244	0.2409	0.1283
0.0577	0.1061	0.1271	0.0639	0.0377	0.0689	0.0830	0.0339
0.0658	0.1297	0.1560	0.0777	0.0587	0.1115	0.1307	0.0622
0.0392	0.0774	0.0971	0.0432	0.0390	0.0851	0.0985	0.0417
0.0420	0.0812	0.1014	0.0445	0.0217	0.0452	0.0573	0.0214
0.0552	0.1091	0.1398	0.0671	0.1372	0.2373	0.2519	0.1295
0.0634	0.1384	0.1592	0.0789	0.0716	0.1323	0.1405	0.0661
0.1458	0.2665	0.3325	0.2298	0.0505	0.1088	0.1285	0.0633
0.0650	0.1212	0.1381	0.0612	0.0763	0.1502	0.1766	0.0927
0.0971	0.1815	0.1887	0.0959	0.0612	0.1171	0.1333	0.0622
0.0580	0.1168	0.1378	0.0607	0.0424	0.0846	0.1079	0.0493
0.0507	0.1067	0.1275	0.0584	0.0454	0.0893	0.1189	0.0572
0.0437	0.0901	0.1031	0.0425	0.0475	0.0987	0.1113	0.0478
0.0398	0.0790	0.0937	0.0391	0.0460	0.0908	0.1046	0.0423
0.0429	0.0900	0.1005	0.0421	0.0346	0.0698	0.0820	0.0312
0.0313	0.0602	0.0817	0.0368	0.0648	0.1230	0.1523	0.0722
0.0415	0.0835	0.1005	0.0467	0.0661	0.1214	0.1205	0.0543
0.0998	0.1682	0.1662	0.0804	0.0316	0.0641	0.0749	0.0284
0.0509	0.1067	0.1296	0.0613	0.0905	0.1633	0.1584	0.0759
0.0692	0.1388	0.1636	0.0830	0.0428	0.0912	0.1041	0.0433
0.1630	0.2862	0.2995	0.1605	0.0420	0.0931	0.1159	0.0540
0.0468	0.1044	0.1364	0.0648	0.0485	0.0977	0.1125	0.0493
0.0327	0.0624	0.0753	0.0293	0.0561	0.1134	0.1261	0.0562
0.0911	0.1657	0.1669	0.0801	0.0461	0.1011	0.1183	0.0568
0.0438	0.0843	0.1025	0.0438	0.0319	0.0626	0.0827	0.0354
0.0967	0.1735	0.1869	0.0924	0.0339	0.0706	0.0931	0.0407
0.0659	0.1036	0.0730	0.0000	0.0335	0.0679	0.0794	0.0307

0.0425	0.0818	0.0976	0.0408	0.0309	0.0646	0.0769	0.0301
0.0477	0.1039	0.1303	0.0609	0.0417	0.0862	0.1010	0.0441
0.0245	0.0516	0.0628	0.0246	0.0375	0.0706	0.0914	0.0397
0.0393	0.0931	0.1095	0.0547	0.0507	0.0992	0.1145	0.0486
0.0532	0.1096	0.1216	0.0543	0.0431	0.0746	0.0916	0.0374
0.1027	0.1937	0.1882	0.0866	0.0461	0.0920	0.1153	0.0536
0.0361	0.0788	0.0977	0.0428	0.0427	0.0872	0.1003	0.0434
0.0342	0.0697	0.0852	0.0360	0.0411	0.0917	0.1045	0.0442
0.0391	0.0826	0.1034	0.0454	0.0554	0.1147	0.1279	0.0596
0.0482	0.1057	0.1256	0.0564	0.0350	0.0767	0.0876	0.0344
0.0484	0.1013	0.1296	0.0594	0.0395	0.0880	0.1046	0.0454
0.0497	0.0945	0.1111	0.0562	0.0679	0.1313	0.1313	0.0594
0.0935	0.1770	0.1847	0.1010	0.0726	0.1505	0.1577	0.0763
0.0515	0.0990	0.1250	0.0540	0.0793	0.1816	0.1990	0.0931
0.0627	0.1256	0.1280	0.0577	0.0639	0.1116	0.1203	0.0528
0.1281	0.2626	0.3122	0.1813	0.0394	0.0775	0.0905	0.0363
0.0280	0.0544	0.0718	0.0294	0.0360	0.0759	0.0959	0.0405
0.0434	0.0849	0.1153	0.0544	0.0346	0.0695	0.0853	0.0321
0.0706	0.1375	0.1537	0.0728	0.0322	0.0641	0.0781	0.0289
0.0710	0.1411	0.1524	0.0725	0.0480	0.0986	0.1132	0.0468
0.0326	0.0680	0.0820	0.0319	0.1337	0.2258	0.2392	0.1219
0.0865	0.1584	0.1594	0.0777	0.0467	0.0954	0.1091	0.0446
0.0336	0.0712	0.0855	0.0362	0.0436	0.0930	0.1059	0.0430
0.0798	0.1548	0.1625	0.0845	0.0830	0.1710	0.1805	0.0857
0.0335	0.0800	0.1035	0.0472	0.0738	0.1571	0.1715	0.0843
0.0252	0.0574	0.0705	0.0297	0.0512	0.0987	0.1107	0.0453
0.0410	0.0872	0.1034	0.0432	0.0412	0.0840	0.0935	0.0374
0.0660	0.1219	0.1226	0.0583	0.0702	0.1332	0.1332	0.0611
0.0460	0.0901	0.1135	0.0515	0.0367	0.0783	0.0850	0.0320
0.0354	0.0717	0.0861	0.0374	0.0563	0.1079	0.1179	0.0549
0.0451	0.0915	0.1042	0.0461	0.0267	0.0154	0.0000	0.0000
0.0628	0.1229	0.1255	0.0584	0.0790	0.1703	0.1857	0.0938
0.0285	0.0629	0.0744	0.0309	0.0554	0.1155	0.1210	0.0553
0.0237	0.0697	0.0817	0.0382	0.1970	0.3357	0.3319	0.1789
0.0433	0.0913	0.1031	0.0482	0.0037	0.0015	0.0000	0.0000
0.0491	0.1054	0.1419	0.0633	0.0367	0.0771	0.0849	0.0324
0.0502	0.1078	0.1290	0.0609	0.0000	0.0023	0.0000	0.0000
0.0414	0.0954	0.1201	0.0580	0.0446	0.0930	0.1043	0.0408
0.0551	0.1158	0.1361	0.0642	0.0577	0.1140	0.1248	0.0544
0.1013	0.1873	0.1870	0.0913	0.0362	0.0753	0.0902	0.0379
0.0334	0.0817	0.1232	0.0578	0.0363	0.0766	0.0897	0.0371
0.0562	0.1140	0.1452	0.0686	0.0330	0.0725	0.0845	0.0333
0.0507	0.1068	0.1167	0.0511	0.1270	0.2327	0.2286	0.1159
0.0421	0.0889	0.1159	0.0550	0.0550	0.1282	0.1544	0.0739
0.0949	0.1726	0.1837	0.0953	0.0502	0.0985	0.1099	0.0498
0.0554	0.1138	0.1226	0.0561	0.0504	0.1056	0.1096	0.0450
0.0851	0.1689	0.1912	0.0999	0.0447	0.0911	0.1053	0.0448
0.0720	0.1404	0.1444	0.0679	0.0757	0.1478	0.1495	0.0680

0.0451	0.0989	0.1176	0.0515	0.0541	0.0995	0.1003	0.0435
0.0436	0.0866	0.1007	0.0420	0.0000	0.1623	0.0000	0.0000
0.1109	0.1991	0.1967	0.0940	0.0751	0.1350	0.1300	0.0546
0.0402	0.0885	0.1105	0.0499	0.1474	0.2448	0.2277	0.1087
0.0699	0.1411	0.1635	0.0786	0.0798	0.1418	0.1368	0.0579
0.0333	0.0698	0.0816	0.0302	0.1283	0.2153	0.2056	0.0897
0.0585	0.1174	0.1364	0.0644	0.1289	0.2173	0.2064	0.0926
0.0439	0.0920	0.1115	0.0454	0.0800	0.1472	0.1427	0.0621
0.0472	0.1017	0.1240	0.0573	0.0999	0.1865	0.1860	0.0799
0.0376	0.0659	0.0879	0.0388	0.0566	0.1166	0.1356	0.0621
0.0635	0.1187	0.1356	0.0667	0.0394	0.0752	0.0952	0.0427
0.0244	0.0551	0.0801	0.0367	0.0508	0.1052	0.1296	0.0604
0.0656	0.1402	0.1535	0.0737	0.0432	0.0861	0.1003	0.0462
0.0497	0.1113	0.1362	0.0630	0.1620	0.2804	0.2942	0.1500
0.0435	0.0955	0.1134	0.0492	0.0783	0.1494	0.1600	0.0777
0.0418	0.0894	0.1136	0.0509	0.0473	0.0846	0.1120	0.0558
0.0397	0.0896	0.1192	0.0569	0.0689	0.1297	0.1388	0.0658
0.0404	0.0905	0.1356	0.0761	0.0687	0.1404	0.1618	0.0772
0.0434	0.0972	0.1339	0.0682	0.0514	0.0967	0.1068	0.0459
0.0425	0.0819	0.1047	0.0556	0.0346	0.0729	0.0968	0.0473
0.0277	0.0573	0.0743	0.0309	0.0376	0.0759	0.1192	0.0799
0.0353	0.0779	0.0903	0.0378	0.0545	0.1296	0.1679	0.0887
0.0518	0.1109	0.1290	0.0569	0.0298	0.0566	0.0721	0.0257
0.0657	0.1430	0.1599	0.0753	0.0427	0.0902	0.1065	0.0445
0.0654	0.1330	0.1608	0.0767	0.1075	0.2036	0.2077	0.1072
0.0453	0.0953	0.1125	0.0482	0.0420	0.0817	0.1040	0.0435
0.0659	0.1310	0.1407	0.0665	0.0000	0.0000	0.1608	0.0000
0.0530	0.1150	0.1302	0.0585	0.0397	0.0758	0.0864	0.0335
0.0673	0.1355	0.1411	0.0670	0.0524	0.1259	0.1407	0.0710
0.0375	0.0759	0.0979	0.0473	0.0468	0.0997	0.1205	0.0558
0.0481	0.1158	0.1394	0.0653	0.1208	0.2270	0.2327	0.1038
0.1104	0.1996	0.1879	0.0869	0.0366	0.0797	0.0902	0.0357
0.0485	0.0995	0.1196	0.0492	0.0776	0.1490	0.1603	0.0783
0.1476	0.2435	0.2299	0.1112	0.0571	0.1151	0.1388	0.0670
0.1199	0.2128	0.1972	0.0878	0.0196	0.0405	0.0516	0.0193
0.0909	0.1420	0.1820	0.1146	0.0462	0.0937	0.1104	0.0516
0.0679	0.1385	0.1724	0.0918	0.0492	0.0978	0.1096	0.0483
0.0417	0.0905	0.1061	0.0478	0.0702	0.1521	0.1738	0.0825
0.0460	0.0939	0.1035	0.0421	0.0286	0.0483	0.0637	0.0244
0.0497	0.1038	0.1398	0.0652	0.0567	0.1078	0.1175	0.0546
0.0785	0.1437	0.1530	0.0699	0.0943	0.1756	0.2049	0.1098
0.0416	0.0901	0.1091	0.0469	0.0350	0.0635	0.0755	0.0284
0.0325	0.0700	0.0931	0.0441	0.0574	0.1177	0.1526	0.0777
0.0880	0.1749	0.2022	0.1128	0.0524	0.1151	0.1597	0.0802
0.0434	0.0851	0.1075	0.0523	0.0745	0.1779	0.1876	0.0813
0.0539	0.0996	0.1163	0.0489	0.0406	0.0708	0.0924	0.0378
0.1124	0.1972	0.2076	0.1052	0.0353	0.0716	0.0864	0.0366
0.0548	0.1157	0.1291	0.0582	0.0082	0.0178	0.0299	0.0116

0.0701	0.1319	0.1352	0.0633	0.1386	0.2358	0.2542	0.1360
0.0739	0.1528	0.1919	0.0865	0.0131	0.0258	0.0369	0.0139
0.0541	0.1131	0.1389	0.0627	0.0541	0.1066	0.1139	0.0524
0.0672	0.1307	0.1434	0.0689	0.1339	0.2363	0.2247	0.1125
0.1037	0.1745	0.1874	0.0975	0.1626	0.2634	0.2976	0.1702
0.1315	0.2254	0.2404	0.1173	0.1143	0.2102	0.2067	0.1003
0.0447	0.0948	0.1343	0.0712	0.0882	0.1699	0.1822	0.0991

TABLE B.4: The local bipartite clustering coefficients of users who were the first to rate a new story.

B.5 Non-English movies in the MovieLens network

Movie title	actual number of ratings (average rating)	predicted number of ratings (predicted average rating)	Rotten Tomatoes (average)	Metacritic (average)
Red Lights (Feux rouges) (2004)	3 (2.88)	19 (2.17)	86 (83%)	28 (74)
Lost Embrace (El Abrazo Partido) (2004)	2 (3)	17 (1.59)	48 (83%)	23 (70)
Sea Inside, The (Mar adentro) (2004)	7 (4.11)	19 (2.66)	131 (84%)	38 (74)
Machuca (2004)	7 (2.28)	21 (3.08)	33 (89%)	37 (76)
Tae Guk Gi - The Brotherhood of War (Taegukgi hwinalrimyeo) (2004)	5 (4.06)	37 (4.22)	41 (80%)	19 (64)
Appleseed (Appurushīdo) (2004)	7 (4.21)	22 (2.91)	32 (25%)	17 (40)
Turtles Can Fly (Lakposhthâ ham parvaz mikonand) (2004)	2 (3.5)	12 (2.02)	72 (88%)	31 (85)
Loop the Loop (a.k.a. Up and Down) (Horem pádem) (2004)	2 (4.25)	40 (4.49)	65 (83%)	27 (78)
Walk on Water (2004)	3 (4.25)	18 (3.13)	75 (72%)	28 (65)
Look at Me (Comme une image) (2004)	5 (3.31)	27 (4.17)	98 (87%)	30 (79)
Year of the Yao, The (2004)	1 (4)	17 (2.6)	33 (67%)	11 (62)
Three... Extremes (Saam gaang yi) (2004)	8 (3.84)	34 (4.38)	62 (84%)	22 (66)
Bittersweet Life, A (Dalkomhan insaeng) (2005)	4 (3.25)	17 (2.35)	10 (100%)	na
Duck Season (Temporada de patos) (2004)	5 (3.38)	19 (2.92)	73 (90%)	27 (74)
Usphizin (2004)	7 (3.91)	17 (2.85)	61 (93%)	na
Vinci (2004)	2 (3.5)	29 (3.97)	na	na

Business, The (2005)	3 (3.63)	20 (2.48)	na	na
Tony Takitani (2004)	3 (3.38)	22 (2.65)	52 (88%)	22 (88)
Child, The (L'Enfant) (2005)	6 (3.94)	20 (3.26)	na	34 (87)
Hidden Blade, The (Kakushi ken oni no tsume) (2004)	4 (3.75)	29 (4.01)	31 (87%)	11 (76)
Three Times (Zui Hao De Shi Guang) (2005)	4 (3.06)	19 (2.68)	50 (86%)	22 (80)
Taxidermia (2006)	5 (4.16)	19 (2.4)	46 (80%)	9 (83)
Gui Si (Silk) (2006)	1 (3.5)	12 (2.26)	5 (40%)	na
Arn - The Knight Templar (Arn - Tempelriddaren) (2007)	3 (3.5)	16 (2.56)	na	na
Tell No One (Ne le dis a personne) (2007)	8 (3.71)	34 (4.88)	108 (94%)	30 (82)
Czech Dream (Český sen) (2004)	2 (3.75)	19 (2.4)	24 (79%)	7 (72)
4 Months, 3 Weeks and 2 Days (4 luni, 3 săptămâni şi 2 zile) (2007)	35 (3.82)	46 (4.5)	133 (95%)	37 (97)
Om Shanti Om (2007)	3 (3.88)	25 (2.94)	13 (77%)	na
Aerial, The (La Antena) (2007)	3 (4)	48 (4.99)	11 (64%)	na
Inside (À l'intérieur) (2007)	5 (3.13)	17 (2.35)	12 (83%)	na
Unknown Solider, The (Unbekannte Soldat, Der) (2006)	1 (3)	13 (2.02)	10 (60%)	6 (71)
Aleksandra (2007)	1 (3)	13 (2.02)	na	13 (85)
Ganes (2007)	2 (3)	20 (2.22)	na	na
Katyn (2007)	4 (3.63)	21 (1.8)	64 (94%)	17 (81)
Maria Full of Grace (Maria, Llena eres de gracia) (2004)	7 (4)	19 (2.44)	139 (97%)	39 (87)
Veer Zaara (2004)	2 (3.25)	40 (4.99)	na	5 (67)
Bad Education (La Mala educación) (2004)	15 (3.79)	30 (4.58)	137 (88%)	34 (81)

TABLE B.5: The table lists the non-English movies that were predicted to receive a higher than the actual number of ratings. In general, these movies received very positive reviews from critics. The low number of ratings received by MovieLens users may be due to their demographics. We listed the number of ratings that were recorded by the websites Rotten Tomatoes and Metacritic. In addition, for the website Rotten Tomatoes the table displays the tomatometer score that represents the percentage of approved critics that have given the movie a positive review. For Metacritic we also show the metascore. The metascore ranges between 0 and 100, with 100 being the best possible score.

Bibliography

[1] Asratian, A. S., Denley, T. M., and Häggkvist, R. *Bipartite Graphs and their Applications.* Cambridge University Press, Cambridge, 1998.

[2] Australian Crime Commission. Illicit drug data report. `https://www.crimecommission.gov.au/sites/default/files/290414-IDDR-2012-13.pdf`, Last accessed: 11-12-2015.

[3] Australian Institute of Criminology. Property crime, 2014. `http://www.aic.gov.au/crime_types/propertycrime.html`, Last accessed: 04-08-2016.

[4] Australian Institute of Criminology. Homicide statistics, 2015. `http://www.aic.gov.au/statistics/homicide.html`, Last accessed: 11-12-2015.

[5] Australian Institute of Family Studies. Child abuse and neglect statistics, 2015. `https://aifs.gov.au/cfca/publications/child-abuse-and-neglect-statistics`, Last accessed: 11-12-2015.

[6] Barabási, A.-L. and Albert, R. Emergence of scaling in random networks. *Science*, 286(5439):509–512, 1999.

[7] Barbera, P., Piccirilli, M., and Geisler, A. Rfacebook: Access to Facebook API via R, 2016. `https://cran.r-project.org/web/packages/Rfacebook/index.html`, Last accessed: 02-09-2016.

[8] Barnes, F. Zell Miller endorses Bush, 2003. `http://www.weeklystandard.com/zell-miller-endorses-bush/article/4550`, Last accessed: 26-07-2016.

[9] Barrat, A., Barthélemy, M., Pastor-Satorras, R., and Vespignani, A. The architecture of complex weighted networks. *Proceedings of the National Academy of Sciences*, 101(11):3747–3752, 2004.

[10] BBC News. Obituary: Noordin Mohamed Top, 2009. `http://news.bbc.co.uk/2/hi/asia-pacific/4302368.stm`, Last accessed: 20-04-2014.

[11] Bercovitz, B. Viewing implicit social networks as bipartite graphs. `http://snap.stanford.edu/class/cs224w-2010/proj2009/finalpaper_Bercovitz.pdf`, Last accessed: 13-05-2016.

[12] Blondel, V. D., Guillaume, J.-L., Lambiotte, R., and Lefebvre, E. Fast unfolding of communities in large networks. *Journal of Statistical Mechanics*, 2008(10):P10008, 2008.

[13] Bonacich, P. Using Boolean algebra to analyze overlapping memberships. *Sociological Methodology*, 9:101–115, 1978.

[14] Borgatti, S. P. and Everett, M. G. Network analysis of 2-mode data. *Social Networks*, 19(3):243–269, 1997.

[15] Breiger, R. L. The duality of persons and groups. *Social Forces*, 53(2):181–190, 1974.

[16] Caldarelli, G., Battiston, S., Garlaschelli, D., and Catanzaro, M. Emergence of complexity in financial networks. In *Complex Networks*, Lecture Notes in Physics, pages 399–423. Springer Berlin Heidelberg, Berlin, Heidelberg, 2004.

[17] Camina, A. and Lewis, B. *An Introduction to Enumeration*. Springer Undergraduate Mathematics Series. Springer London, London, 2011.

[18] Carstens, C. J. Proof of uniform sampling of binary matrices with fixed row sums and column sums for the fast Curveball algorithm. *Physical Review E*, 91(4):042812, 2015.

[19] Chan, H. S. and Dill, K. A. The protein folding problem. *Physics today*, 46(2):24–32, 1993.

[20] Chen, D.-B., Gao, H., Lü, L., and Zhou, T. Identifying influential nodes in large-scale directed networks: The role of clustering. *PLoS ONE*, 8(10):e77455, 2013.

[21] Chen, D.-B., Xiao, R., Zeng, A., and Zhang, Y.-C. Path diversity improves the identification of influential spreaders. *EPL (Europhysics Letters)*, 104(6):68006, 2013.

[22] Chen, D., Lü, L., Shang, M.-S., Zhang, Y.-C., and Zhou, T. Identifying influential nodes in complex networks. *Physica A*, 391(4):1777–1787, 2012.

[23] Clauset, A., Newman, M. E. J., and Moore, C. Finding community structure in very large networks. *Physical Review E*, 70(6):066111, 2004.

[24] Csárdi, G. and Nepusz, T. The igraph software package for complex network research. *InterJournal, Complex Systems*, 1695(5):1–9, 2006.

[25] Cui, Y., Wang, X., and Li, J. Detecting overlapping communities in networks using the maximal sub-graph and the clustering coefficient. *Physica A*, 405:85–91, 2014.

[26] Davies, T. and Marchione, E. Event networks and the identification of crime pattern motifs. *PLoS ONE*, 10(11):e0143638, 2015.

[27] Davis, A., Gardner, B. B., and Gardner, M. R. *Deep South: A Social Anthropological Study of Caste and Class.* University of Chicago Press, Chicago, 1941.

[28] Diestel, R. *Graph Theory.* Springer, New York, 2005.

[29] Doreian, P. On the delineation of small group structure. In *Classifying Social Data.* Jossey-Bass, San Francisco, 1979.

[30] D'Orsogna, M. R. and Perc, M. Statistical physics of crime: A review. *Physics of Life Reviews*, 12:1–21, 2015.

[31] Easton, C., Swan, S., and Sinha, R. Motivation to change substance use among offenders of domestic violence. *Journal of Substance Abuse Treatment*, 19(1):1–5, 2000.

[32] Euler, L. Solutio problematis ad geometriam situs pertinentis. *Commentarii academiae scientiarum Petropolitanae*, 8:128–140, 1741.

[33] Everett, M. G. and Borgatti, S. P. The dual-projection approach for two-mode networks. *Social Networks*, 35(2):204–210, 2013.

[34] Everett, M. G. and Borgatti, S. P. An extension of regular colouring of graphs to digraphs, networks and hypergraphs. *Social Networks*, 15(3):237–254, 1993.

[35] Everton, S. *Disrupting Dark Networks.* Cambridge University Press, Cambridge, 2012.

[36] Feld, S. L. Why your friends have more friends than you do. *American Journal of Sociology*, 96(6):1464–1477, 1991.

[37] Ferrer, R. and Solé, R. V. The small world of human language. *Proceedings of the Royal Society B*, 268(1482):2261–2265, 2001.

[38] Forbes, C., Evans, M., Hastings, N., and Peacock, B. *Statistical Distributions.* John Wiley & Sons, Inc., Hoboken, 2011.

[39] Foti, N. J., Hughes, J. M., and Rockmore, D. N. Nonparametric sparsification of complex multiscale networks. *PLoS ONE*, 6(2):e16431, 2011.

[40] Fowler, J. H. Connecting the Congress: A study of cosponsorship networks. *Political Analysis*, 14(4):456–487, 2006.

[41] Fowler, J. H. Legislative cosponsorship networks in the US House and Senate. *Social Networks*, 28(4):454–465, 2006.

[42] Girvan, M. and Newman, M. E. J. Community structure in social and biological networks. *Proceedings of the National Academy of Sciences*, 99(12):7821–7826, 2002.

[43] Glattfelder, J. B. and Battiston, S. Backbone of complex networks of corporations: The flow of control. *Physical Review E*, 80(3):036104, 2009.

[44] Grossman, J. W. and Ion, P. D. F. On a portion of the well-known collaboration graph. *Congressus Numerantium*, pages 129–132, 1995.

[45] Guillaume, J.-L. and Latapy, M. Bipartite structure of all complex networks. *Information Processing Letters*, 90(5):215–221, 2004.

[46] Guttmann, A. J. Self-avoiding walks and polygons – an overview. *arXiv:1212.3448v1*, pages 1–19, 2012.

[47] Harper, F. M. and Konstan, J. A. The MovieLens Datasets. *ACM Transactions on Interactive Intelligent Systems*, 5(4):19, 2016.

[48] Herlocker, J. L., Konstan, J. A., Terveen, L. G., and Riedl, J. T. Evaluating collaborative filtering recommender systems. *ACM Transactions on Information Systems*, 22(1):5–53, 2004.

[49] Hogg, T. and Lerman, K. Social dynamics of Digg. *EPJ Data Science*, 1(1):5, 2012.

[50] Howard, M. Malicious damage to property offences in New South Wales, 2006. `http://www.austlii.edu.au/au/journals/NSWCrimJustB/2006/11.pdf`, Last accessed: 06-08-2016.

[51] Humphreys, K. A history and a survey of lattice path enumeration. *Journal of Statistical Planning and Inference*, 140(8):2237–2254, 2010.

[52] Johnson, S. D., Bowers, K., and Hirschfield, A. New insights into the spatial and temporal distribution of repeat victimization. *British Journal of Criminology*, 37 (2):224–241, 1997.

[53] Kashani, Z., Ahrabian, H., Elahi, E., Nowzari-Dalini, A., Ansari, E., Asadi, S., Mohammadi, S., Schreiber, F., and Masoudi-Nejad, A. Kavosh: a new algorithm for finding network motifs. *BMC Bioinformatics*, 10(1):318, 2009.

[54] Khakabimamaghani, S., Sharafuddin, I., Dichter, N., Koch, I., and Masoudi-Nejad, A. QuateXelero: An accelerated exact network motif detection algorithm. *PLoS ONE*, 8(7):e68073, 2013.

[55] Kitsak, M., Gallos, L. K., Havlin, S., Liljeros, F., Muchnik, L., Stanley, H. E., and Makse, H. A. Identification of influential spreaders in complex networks. *Nature Physics*, 6(11):888–893, 2010.

[56] Kogut, B. and Walker, G. The small world of Germany and the durability of national networks. *American Sociological Review*, 66(3):317–335, 2001.

[57] KONECT. Networks, 2014. `http://konect.uni-koblenz.de/networks/`, Last accessed: 31-08-2016.

[58] Lancichinetti, A. and Fortunato, S. Community detection algorithms: A comparative analysis. *Physical Review E*, 80(5):056117, 2009.

[59] Lancichinetti, A. and Fortunato, S. Limits of modularity maximization in community detection. *Physical Review E*, 84(6):066122, 2011.

[60] Lando, S. K. *Lectures on Generating Functions*. American Mathematical Society, Providence, 2003.

[61] Latapy, M., Magnien, C., and Vecchio, N. D. Basic notions for the analysis of large two-mode networks. *Social Networks*, 30(1):31–48, 2008.

[62] Le Cam, L. An approximation theorem for the Poisson binomial distribution. *Pacific Journal of Mathematics*, 10(4):1181–1197, 1960.

[63] Li, J. and Zhou, J. Chinese character structure analysis based on complex networks. *Physica A*, 380(1):629–638, 2007.

[64] Li, M., Fan, Y., Chen, J., Gao, L., Di, Z., and Wu, J. Weighted networks of scientific communication: The measurement and topological role of weight. *Physica A*, 350 (2-4):643–656, 2005.

[65] Li, Q., Zhou, T., Lü, L., and Chen, D. Identifying influential spreaders by weighted LeaderRank. *Physica A*, 404:47–55, 2014.

[66] Liebig, J. and Rao, A. Identifying influential nodes in bipartite networks using the clustering coefficient. In *2014 Tenth International Conference on Signal-Image Technology and Internet-Based Systems*, pages 323–330. IEEE, 2014.

[67] Liebig, J. and Rao, A. Predicting item popularity: Analysing local clustering behaviour of users. *Physica A*, 442:523–531, 2016.

[68] Liebig, J. and Rao, A. The case study of an Australian crime dataset. In *The Third Annual Conference of Research@Locate*, pages 30–35, 2016.

[69] Liebig, J. and Rao, A. Fast extraction of the backbone of projected bipartite networks to aid community detection. *EPL (Europhysics Letters)*, 113(2):28003, 2016.

[70] Lind, P. G., González, M. C., and Herrmann, H. J. Cycles and clustering in bipartite networks. *Physical Review E*, 72(5):056127, 2005.

[71] Lü, L., Zhang, Y.-C., Yeung, C. H., and Zhou, T. Leaders in social networks, the Delicious case. *PLoS ONE*, 6(6):e21202, 2011.

[72] Mane, K. K. and Börner, K. Mapping topics and topic bursts in PNAS. *Proceedings of the National Academy of Sciences*, 101:5287–5290, 2004.

[73] McKay, B. D. Practical graph isomorphism. *Congressus Numerantium*, 30:45–87, 1981.

[74] Medo, M., Cimini, G., and Gualdi, S. Temporal effects in the growth of networks. *Physical Review Letters*, 107(23):238701, 2011.

[75] Milgram, S. The small world problem. *Psychology Today*, 1(1):61–67, 1967.

[76] Milo, R., Kashtan, N., Itzkovitz, S., Newman, M. E. J., and Alon, U. On the uniform generation of random graphs with prescribed degree sequences. *arXiv:cond-mat/0312028v2*, pages 1–4, 2004.

[77] Milo, R., Shen-Orr, S., Itzkovitz, S., Kashtan, N., Chklovskii, D., and Alon, U. Network motifs: Simple building blocks of complex networks. *Science*, 298(5594): 824–827, 2002.

[78] Montgomery, L. Kent Conrad's last stand on debt, 2012. `https://www.washingtonpost.com/business/economy/kent-conrads-last-stand-on-debt/2012/03/08/gIQABudYGS{_}story.html`, Last accessed: 26-07-2016.

[79] Nadakuditi, R. R. and Newman, M. E. J. Graph spectra and the detectability of community structure in networks. *Physical Review Letters*, 108(18):188701, 2012.

[80] Nagler, J. Scobit: An alternative estimator to Logit and Probit. *American Journal of Political Science*, 38(1):230–255, 1994.

[81] Neal, Z. Identifying statistically significant edges in one-mode projections. *Social Network Analysis and Mining*, 3(4):915–924, 2013.

[82] Neal, Z. The backbone of bipartite projections: Inferring relationships from co-authorship, co-sponsorship, co-attendance and other co-behaviors. *Social Networks*, 39:84–97, 2014.

[83] Newman, M. E. J. Properties of highly clustered networks. *Physical Review E*, 68 (2):026121, 2003.

[84] Newman, M. E. J. The structure and function of complex networks. *SIAM Review*, 45(2):167–256, 2003.

[85] Newman, M. E. J. Finding community structure in networks using the eigenvectors of matrices. *Physical Review E*, 74(3):036104, 2006.

[86] Newman, M. E. J. *Networks: An Introduction*. Oxford University Press, Oxford, 2010.

[87] Newman, M. E. J. and Girvan, M. Finding and evaluating community structure in networks. *Physical Review E*, 69(2):026113, 2004.

[88] Newman, M. E. J. and Peixoto, T. P. Generalized communities in networks. *Physical Review Letters*, 115(8):088701, 2015.

[89] Newman, M. E. J. The structure of scientific collaboration networks. *Proceedings of the National Academy of Sciences*, 98(2):404–409, 2001.

[90] Newman, M. E. J. Scientific collaboration networks. I. Network construction and fundamental results. *Physical Review E*, 64(1):016131, 2001.

[91] Newman, M. E. J. Scientific collaboration networks. II. Shortest paths, weighted networks, and centrality. *Physical Review E*, 64(1):016132, 2001.

[92] Oliveira, M., Barbosa-Filho, H., Yehle, T., White, S., and Menezes, R. From criminal spheres of familiarity to crime networks. In *Complex Networks VI*, volume 597, pages 219–230. 2015.

[93] Omidi, S., Schreiber, F., and Masoudi-Nejad, A. MODA: An efficient algorithm for network motif discovery in biological networks. *Genes & Genetic Systems*, 84 (5):385–395, 2009.

[94] Opsahl, T. Triadic closure in two-mode networks: Redefining the global and local clustering coefficients. *Social Networks*, 35(2):159–167, 2013.

[95] Page, L., Brin, S., Motwani, R., and Winograd, T. The PageRank citation ranking: Bringing order to the web. *Stanford InfoLab*, 1999.

[96] Paoletti, T. Leonard Euler's solution to the Königsberg bridge problem, 2011. `http://www.maa.org/press/periodicals/convergence/leonard-eulers-solution-to-the-konigsberg-bridge-problem`, Last accessed: 20-06-2016.

[97] Partridge, E. Porn and domestic violence: NSW Police says respect for women from young men crucial, 2014. `http://www.smh.com.au/nsw/porn-and-domestic-violence-nsw-police-says-respect-for-women-from-young-men-crucial-20141204-1205hy.html`, Last accessed: 04-08-2016.

[98] Pastor-Satorras, R. and Vespignani, A. Immunization of complex networks. *Physical Review E*, 65(3):036104, 2002.

[99] Peixoto, T. P. Eigenvalue Spectra of Modular Networks. *Physical Review Letters*, 111(9):098701, 2013.

[100] Phillips, A. A GOP senator might vote for Hillary Clinton. Here's how rare that is, 2016. `https://www.washingtonpost.com/news/the-fix/wp/2016/06/11/a-gop-senator-might-vote-for-hillary-clinton-heres-how-rare-that-is/`, Last accessed: 26-07-2016.

[101] Pons, P. and Latapy, M. Computing communities in large networks using random walks. In *Computer and Information Sciences-ISCIS*, volume 3733, pages 284–293. 2005.

[102] Price, D. D. S. A general theory of bibliometric and other cumulative advantage processes. *Journal of the American Society for Information Science and Technology*, 27(5):292–306, 1976.

[103] R Core Team. R: A language and environment for statistical computing, 2016. `https://www.r-project.org/`, Last accessed: 31-08-2016.

[104] Raghavan, U. N., Albert, R., and Kumara, S. Near linear time algorithm to detect community structures in large-scale networks. *Physical Review E*, 76(3):036106, 2007.

[105] Ravasz, E., Somera, A., Mongru, D., Oltavi, Z., and Barabási, A.-L. Hierarchical organization of modularity in metabolic networks. *Science*, 297(5586):1551–1555, 2002.

[106] Resnick, P., Iacovou, N., Suchak, M., Bergstrom, P., and Riedl, J. GroupLens: An open architecture for collaborative filtering of netnews. In *Proceedings of the 1994 ACM conference on Computer supported cooperative work - CSCW '94*, pages 175–186, New York, New York, USA, 1994. ACM Press.

[107] Reuters. FACTBOX: Five facts about Islamic militant Noordin Top, 2009. `http://www.reuters.com/article/us-indonesia-militants-top-sb-idUSTRE58G1HE20090917`, Last accessed: 23-08-2016.

[108] Ribeiro, P. and Silva, F. G-Tries: A data structure for storing and finding subgraphs. *Data Mining and Knowledge Discovery*, 28(2):337–377, 2014.

[109] Robins, G. and Alexander, M. Small worlds among interlocking directors: Network structure and distance in bipartite graphs. *Computational & Mathematical Organization Theory*, 10(1):69–94, 2004.

[110] Rosvall, M. and Bergstrom, C. T. Maps of random walks on complex networks reveal community structure. *Proceedings of the National Academy of Sciences*, 105 (4):1118–1123, 2008.

[111] Rosvall, M., Axelsson, D., and Bergstrom, C. T. The map equation. *The European Physical Journal Special Topics*, 178(1):13–23, 2009.

[112] Schafer, J. B., Frankowski, D., Herlocker, J., and Sen, S. Collaborative filtering recommender systems. In *The Adaptive Web*, pages 291–324. Springer Berlin Heidelberg, Berlin, Heidelberg, 2007.

[113] Scutari, M. and Nagarajan, R. Identifying significant edges in graphical models of molecular networks. *Artificial Intelligence in Medicine*, 57(3):207–217, 2013.

[114] Serrano, M. Á., Boguna, M., and Vespignani, A. Extracting the multiscale backbone of complex weighted networks. *Proceedings of the National Academy of Sciences*, 106(16):6483–6488, 2009.

[115] Shen-Orr, S. S., Milo, R., Mangan, S., and Alon, U. Network motifs in the transcriptional regulation network of Escherichia coli. *Nature Genetics*, 31(1):64–68, 2002.

[116] Sherrill, M. Maine senators may not like each other much, but they share love of state, job, 2011. `https://www.washingtonpost.com/lifestyle/style/maine-senators-may-not-like-each-other-much-but-they-share-love-of-state-job/2011/05/03/AFGwpnOF{_}story.html`, Last accessed: 26-07-2016.

[117] Short, M. B., D'Orsogna, M. R., Pasour, V. B., Tita, G. E., Brantingham, P. J., Bertozzi, A. L., and Chayes, L. B. A statistical model of criminal behavior. *Mathematical Models and Methods in Applied Sciences*, 18:1249–1267, 2008.

[118] Slater, P. B. A two-stage algorithm for extracting the multiscale backbone of complex weighted networks. *Proceedings of the National Academy of Sciences*, 106 (26):E66, 2009.

[119] Strona, G., Nappo, D., Boccacci, F., Fattorini, S., and San-Miguel-Ayanz, J. A fast and unbiased procedure to randomize ecological binary matrices with fixed row and column totals. *Nature Communications*, 5:4114, 2014.

[120] Sullivan, S. A brief history of Lincoln Chafee's party identity crises, 2013. `https://www.washingtonpost.com/news/the-fix/wp/2013/05/30/a-brief-history-of-lincoln-chafees-party-identity-crises/`, Last accessed: 26-07-2016.

[121] Sweeney, J. and Payne, J. Alcohol and disorderly conduct on Friday and Saturday nights, Australian Institute of Criminology, 2011. `http://www.aic.gov.au/publications/current%20series/rip/1-10/15.html`, Last accessed: 11-12-2015.

[122] Tompson, L., Johnson, S., Ashby, M., Perkins, C., and Edwards, P. UK open source crime data: accuracy and possibilities for research. *Cartography and Geographic Information Science*, 42(2):97–111, 2014.

[123] Ugander, J., Backstrom, L., Marlow, C., and Kleinberg, J. Structural diversity in social contagion. *Proceedings of the National Academy of Sciences*, 109(16): 5962–5966, 2012.

[124] United Nations Office on Drugs and Crime. Drug policy and results in Australia, 2008. `https://www.unodc.org/documents/data-and-analysis/Studies/Drug_Policy_Australia_Oct2008.pdf`, Last accessed: 11-12-2015.

[125] Vig, J., Sen, S., and Riedl, J. The tag genome: Encoding community knowledge to support novel interaction. *ACM Transactions on Interactive Intelligent Systems*, 2 (3):13, 2012.

[126] Vogt, I. and Mestres, J. Drug-target networks. *Molecular Informatics*, 29(1-2): 10–14, 2010.

[127] Wang, Y. H. On the number of successes in independent trials. *Statistica Sinica*, 3(2):295–312, 1993.

[128] Wasserman, S. and Faust, K. *Social Network Analysis: Methods and Applications.* Cambridge University Press, New York, 1994.

[129] Watts, D. J. and Strogatz, S. H. Collective dynamics of 'small-world' networks. *Nature*, 393(6684):440–442, 1998.

[130] Wernicke, S. A faster algorithm for detecting network motifs. In *Algorithms in Bioinformatics*, volume 3692, pages 165–177. 2005.

[131] Wernicke, S. Efficient Detection of Network Motifs. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 3(4):347–359, 2006.

[132] Williams, V. V. Multiplying matrices faster than Coppersmith-Winograd. In *Proceedings of the 44th symposium on Theory of Computing - STOC'12*, pages 887–898, New York, New York, USA, 2012. ACM Press.

[133] Zeng, A., Gualdi, S., Medo, M., and Zhang, Y.-C. Trend prediction in temporal bipartite networks: The case of MovieLens, Netflix, and Digg. *Advances in Complex Systems*, 16(04n05):1350024, 2013.

[134] Zhang, P., Wang, J., Li, X., Li, M., Di, Z., and Fan, Y. Clustering coefficient and community structure of bipartite networks. *Physica A*, 387(27):6869–6875, 2008.

[135] Zhang, X., Zhu, J., Wang, Q., and Zhao, H. Identifying influential nodes in complex networks with community structure. *Knowledge-Based Systems*, 42:74–84, 2013.

[136] Zhang, X., Zhang, Z., Zhao, H., Wang, Q., and Zhu, J. Extracting the globally and locally adaptive backbone of complex networks. *PLoS ONE*, 9(6):e100428, 2014.

[137] Zhou, T., Ren, J., Medo, M., and Zhang, Y.-C. Bipartite network projection and personal recommendation. *Physical Review E*, 76(4):046115, 2007.

[138] Zweig, K. A. and Kaufmann, M. A systematic approach to the one-mode projection of bipartite graphs. *Social Network Analysis and Mining*, 1(3):187–218, 2011.